

# PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2002-342280

(43)Date of publication of application : 29.11.2002

(51)Int.Cl. G06F 15/00  
G06F 9/46  
G06F 13/00  
G06F 15/16

(21)Application number : 2002-057064 (71)Applicant : INTERNATL BUSINESS MACH  
CORP <IBM>

(22)Date of filing : 04.03.2002 (72)Inventor : BASKEY MICHAEL E  
DEGILIO FRANK J  
JONES JOHN C  
ROHRBACH CHRISTIAN F  
TEMPLE JOSEPH L III

(30)Priority

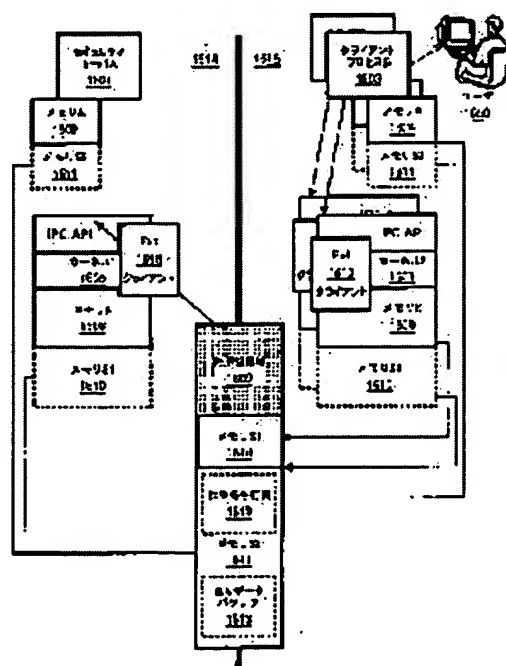
Priority number : 2001 801492 Priority date : 08.03.2001 Priority country : US

(54) PARTITIONED PROCESSING SYSTEM, METHOD FOR SETTING SECURITY IN THE SAME SYSTEM AND COMPUTER PROGRAM THEREOF

(57)Abstract:

PROBLEM TO BE SOLVED: To provide an inter-partition message passing method for a security server in a partitioned processing environment.

SOLUTION: A common security server 1601 is run in a first partition 1614, and at least one security client 1603 is run in at least one other partition 1615, and each partition is provided with a shared memory or inter-memory connection to the first partition, so that a security client server can communicate with the common security server, and a mechanism connected to the security client for transmitting an authorization request by a user to the security client is included. The security



client transmits the authorization request via a main storage area 1609 to the common security server, and the common security server transmits a reply to the authorization request via the main storage area to the security client. Then, the security client transmits the reply to a user 1650.

---

## LEGAL STATUS

[Date of request for examination] 04.03.2002

[Date of sending the examiner's decision of rejection] 07.09.2005

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

(19) 日本国特許庁 (J P)

## (12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2002-342280

(P 2 0 0 2 - 3 4 2 2 8 0 A)

(43) 公開日 平成14年11月29日 (2002. 11. 29)

(51) Int. Cl. <sup>7</sup>	識別記号	F I	テマコード (参考)
G06F 15/00	330	G06F 15/00	330 A 5B045
9/46	350	9/46	350 5B085
13/00	351	13/00	351 Z 5B089
15/16	620	15/16	620 B 5B098

審査請求 有 請求項の数26 O L (全22頁)

(21) 出願番号 特願2002-57064 (P 2002-57064)

(22) 出願日 平成14年3月4日 (2002. 3. 4)

(31) 優先権主張番号 09/801492

(32) 優先日 平成13年3月8日 (2001. 3. 8)

(33) 優先権主張国 米国 (US)

(71) 出願人 390009531

インターナショナル・ビジネス・マシーンズ・コーポレーション

INTERNATIONAL BUSINESS MACHINES CORPORATION

アメリカ合衆国10504、ニューヨーク州  
アーモンク ニュー オーチャード ロード

(74) 代理人 100086243

弁理士 坂口 博 (外1名)

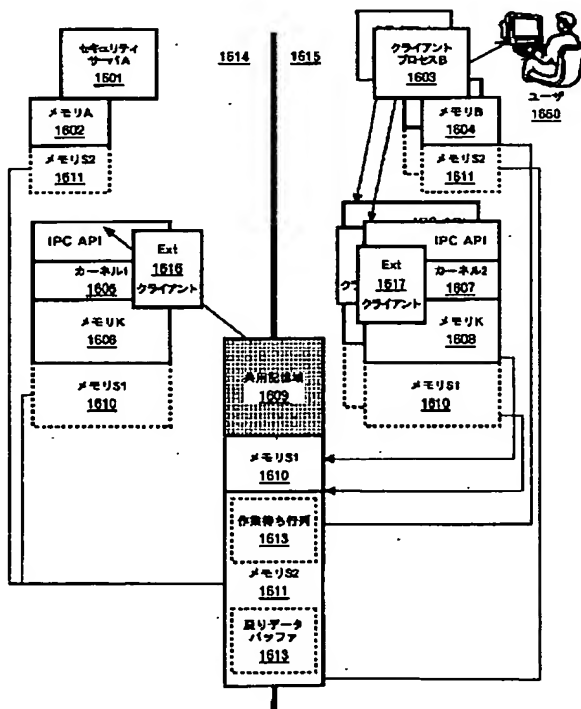
最終頁に続く

(54) 【発明の名称】 区分処理システム、区分処理システムにおけるセキュリティを設ける方法、およびそのコンピュータ・プログラム

(57) 【要約】

【課題】 区分処理環境におけるセキュリティ・サーバのための区分間メッセージ受渡し方法を得る。

【解決手段】 共通セキュリティ・サーバ1601が第1の区分1614で実行され、少なくとも1つのセキュリティ・クライアント1603が、少なくとも1つの他の区分1615で実行され、各区分が共用メモリまたは第1の区分へのメモリ間接続を有し、それによってセキュリティ・クライアントが共通セキュリティ・サーバと通信することができ、セキュリティ・クライアントに接続され、ユーザによる認証要求をセキュリティ・クライアントに送る機構も含む。セキュリティ・クライアントが、認証要求を主記憶域1609を介して共通セキュリティ・サーバに送る。共通セキュリティ・サーバが、認証要求に対する応答を主記憶域を介してセキュリティ・クライアントに送る。次に、セキュリティ・クライアントがその応答をユーザ1650に送る。



## 【特許請求の範囲】

【請求項1】 共通セキュリティ・サーバを含む第1の区分とセキュリティ・クライアントを含む第2の区分とを有する区分処理システムにおいてセキュリティを設ける方法であって、

- a) ユーザによる許可要求を前記第2の区分内の前記セキュリティ・クライアントに送るステップと、
  - b) 前記許可要求を前記セキュリティ・クライアントから前記第1の区分内の前記共通セキュリティ・サーバに送るステップと、
  - c) 前記許可要求に対する第1の応答を前記第1の区分内の前記共通セキュリティ・サーバから前記第2の区分内の前記セキュリティ・クライアントに送るステップとを有し、
- 区分間の前記要求または前記第1の応答のうちのいずれか一方の前記送信が主記憶域を経由し、
- d) さらに、前記セキュリティ・クライアントから前記ユーザに第2の応答を送るステップとを含む方法。

【請求項2】 ステップbの許可要求の送信が、

- b 1) 前記第2の区分内で稼働する前記セキュリティ・クライアントが信号を送って前記第1の区分で稼働している第1のプログラムに前記第1の区分内のプロキシ・クライアントを始動させるステップと、
- b 2) 前記プロキシ・クライアントから前記第1の区分内の前記セキュリティ・サーバに前記要求を送るステップとをさらに含む、請求項1に記載の区分処理システムにおいてセキュリティを設ける方法。

【請求項3】 ステップbまたはステップcのいずれかが、前記第1の区分と前記第2の区分との間で共用される主記憶域を使用するステップを含む、請求項1に記載の区分処理システムにおいてセキュリティを設ける方法。

【請求項4】 ステップbまたはステップcのいずれかが、メモリ間データ移動によって前記第1の区分と前記第2の区分との間にリンクされた主記憶域を使用するステップを含む、請求項1に記載の区分処理システムにおいてセキュリティを設ける方法。

【請求項5】 ステップbが、前記セキュリティ・サーバ固有のインタフェースを使用した前記プロキシ・クライアントによるプログラム呼出しを含む、請求項1に記載の区分処理システムにおいてセキュリティを設ける方法。

【請求項6】 ステップcが、前記共通セキュリティ・サーバから前記セキュリティ・クライアントに前記第1の区分内で稼働している第1のプログラムを介して前記第1の応答を送るステップをさらに含む、請求項1に記載の区分処理システムにおいてセキュリティを設ける方法。

【請求項7】 前記第2の区分内の前記セキュリティ・クライアントから前記ユーザに送られる前記第2の応答が

プログラム処置である、請求項1に記載の区分処理においてセキュリティを設ける方法。

【請求項8】 共通セキュリティ・サーバを含む第1の区分とセキュリティ・クライアントを有する第2の区分とを有するセキュリティを設ける区分処理システムであって、

- ユーザによる許可要求を前記第2の区分内の前記セキュリティ・クライアントに送る手段と、
- 前記許可要求を前記セキュリティ・クライアントから前記第1の区分内の前記共通セキュリティ・サーバに送る手段と、
- 前記許可要求に対する第1の応答を前記第1の区分内の前記共通セキュリティ・サーバから前記第2の区分内の前記セキュリティ・クライアントに送る手段とを有し、
- 区分間の前記要求または前記第1の応答のうちのいずれか一方の前記送信が主記憶域を経由し、
- さらに前記セキュリティ・クライアントから前記ユーザに第2の応答を送る手段とを含む区分処理システム。

【請求項9】 前記許可要求を送る前記手段が、

- 前記第1の区分内で稼働し、プロキシ・クライアントを始動させる第1のプログラムと、
- 前記第2の区分内で稼働している前記セキュリティ・クライアントによって前記第1のプログラムに信号を送り、それによって前記第1の区分内の前記プロキシ・クライアントを始動させる手段と、
- 前記プロキシ・クライアントから前記第1の区分内の前記セキュリティ・サーバに前記要求を送る手段とをさらに含む、請求項8に記載の区分処理システム。

【請求項10】 前記主記憶域が前記第1の区分と前記第2の区分との間で共用される記憶域を含む、請求項8に記載の区分処理システム。

【請求項11】 前記第1の区分と前記第2の区分との間にリンクされた記憶域をさらに含み、前記セキュリティ・クライアントから前記許可要求を送る前記手段がメモリ間データ移動機構を含む、請求項8に記載の区分処理システム。

【請求項12】 前記プロキシ・クライアントから前記要求を送る前記手段が前記セキュリティ・サーバ固有のインタフェースを使用して前記プロキシ・クライアントによるプログラム呼出しを送る手段を含む、請求項8に記載の区分処理システム。

【請求項13】 前記共通セキュリティ・サーバから前記許可要求に対する応答を送る前記手段が、前記第1の区分内で稼働し、前記共通セキュリティ・サーバから前記セキュリティ・クライアントに前記応答を送る第1のプログラムをさらに含む、請求項8に記載の区分処理システム。

【請求項14】 第1の区分が共通セキュリティ・サーバを含み、第2の区分がセキュリティ・クライアントを含む区分処理システムにおいてセキュリティを設けるコン

10

20

30

40

50

ピュータ・プログラムであって、前記システムに、

a) ユーザによる許可要求を前記第2の区分内の前記セキュリティ・クライアントに送る手順と、

b) 前記許可要求を前記セキュリティ・クライアントから前記第1の区分内の前記共通セキュリティ・サーバに送る手順と、

c) 前記許可要求に対する第1の応答を前記第1の区分内の前記共通セキュリティ・サーバから前記第2の区分内の前記セキュリティ・クライアントに送るステップであって、区分間の前記要求または前記第1の応答のうちのいずれか一方の前記送信が主記憶域を経由する手順と、

d) 前記セキュリティ・クライアントから前記ユーザに第2の応答を送る手順とを実現させる、コンピュータ・プログラム。

【請求項15】前記許可要求を送る手順が、

b1) 前記第2の区分内で稼働する前記セキュリティ・クライアントが信号を送って前記第1の区分で稼働している第1のプログラムに前記第1の区分内のプロキシ・クライアントを始動させる手順と、

b2) 前記プロキシ・クライアントから前記第1の区分内の前記セキュリティ・サーバに前記要求を送る手順とをさらに含む、請求項14に記載のコンピュータ・プログラム。

【請求項16】前記手順bまたは手順cにおいて、前記第1の区分と少なくとも1つの前記第2の区分の1つとの間で共用される記憶域を使用する、請求項14に記載のコンピュータ・プログラム製品。

【請求項17】前記手順bまたは手順cにおいて、前記第1の区分と少なくとも1つの前記第2の区分の1つとの間にメモリ欄データ移動機構によってリンクされた記憶域を使用する、請求項14に記載のコンピュータ・プログラム製品。

【請求項18】前記手順bにおいて、前記セキュリティ・サーバ固有のインタフェースを使用して前記プロキシ・クライアントによるプログラム呼出しを供給する、請求項14に記載のコンピュータ・プログラム製品。

【請求項19】前記手順cにおいて、前記共通セキュリティ・サーバから前記セキュリティ・クライアントに前記第1の区分内で稼働している第1のプログラムを介して前記応答を送る、請求項14に記載のコンピュータ・プログラム製品。

【請求項20】共通セキュリティ・サーバを含む第1の区分とセキュリティ・クライアントを有する第2の区分とを有するセキュリティを設ける区分処理システムであって、

前記第1の区分によるアクセスが可能な第1の部分と前記第2の区分によるアクセスが可能な第2の部分とを有する主記憶域と、

前記セキュリティ・クライアントに接続され、ユーザに

よる許可要求を前記セキュリティ・クライアントに送る機構と、

前記セキュリティ・クライアントから前記共通セキュリティ・サーバに前記許可要求を送る第1の送信部と、

前記第1の区分内の前記共通セキュリティ・サーバから前記第2の区分内の前記セキュリティ・クライアントに前記許可要求に対する第1の応答を送る前記共通セキュリティ・サーバ内の第2の送信部と、を有し、前記区分間の前記要求または前記第1の応答のいずれか一方の送信が主記憶域を介し、

さらに、前記セキュリティ・クライアントから前記ユーザに第2の応答を送る前記共通セキュリティ・サーバ内の第3の送信部とを含む、区分処理システム。

【請求項21】前記第1の送信部が、

プロキシ・クライアントを始動させる前記第1の区分内で稼働するプログラムと、

前記プログラムに信号を送り、それによって前記第1の区分内の前記プロキシ・クライアントを始動させる前記セキュリティ・クライアントと、

20 前記プロキシ・クライアントから前記セキュリティ・サーバに前記要求を送る第4の送信部とを含む、請求項20に記載の区分処理システム。

【請求項22】前記主記憶域が、前記第1の区分と前記第2の区分の両方によるアクセスが可能な第3の部分を含む、請求項20に記載の区分処理システム。

【請求項23】前記第1の区分と前記第2の区分の間にリンクされた記憶域をさらに含み、前記第2の送信部が前記主記憶域の前記第1の部分と前記第2の部分との間でデータを移動するメモリ間データ移動機構を含む、請求項20に記載の区分処理システム。

【請求項24】前記第4の送信部が、セキュリティ・サーバ固有のインタフェースを使用して前記プロキシ・クライアントによるプログラム呼出を送る、請求項21に記載の区分処理システム。

【請求項25】前記第2の送信部が、前記第1の区分内で稼働し、前記共通セキュリティ・サーバから前記セキュリティ・クライアントに前記第1の応答を送るプログラムをさらに含む、請求項20に記載の区分処理システム。

40 【請求項26】 第2のセキュリティ・クライアントを有する第3の区分をさらに含み、前記第1の区分の前記共通セキュリティ・サーバが前記第2の区分内の前記セキュリティ・クライアントまたは前記第3の区分内の前記第2のセキュリティ・クライアントからの許可要求に応答する、請求項20に記載の区分処理システム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、一般には区分データ処理システムに関し、例えば、システムの各区分内で複数のオペレーティング・システム・イメージを動作さ

せることが可能な単一プロセッサ・システムおよびマルチプロセッサ・システムに関する。複数のオペレーティング・システムの各オペレーティング・システムは、同種区分処理環境における同一のオペレーティング・システムのイメージとすることができるか、または異種区分処理環境において複数のオペレーティング・システムが複数のオペレーティング・システム・イメージによってサポートされる。

#### 【0002】

【従来の技術】最近の中企業から大企業までの企業のほとんどは、IT基盤を展開させて、従来集中化されていた「ガラス張り」データ・センターのカバー範囲を、組織全体、さらには組織の境界を超えて拡張している。このような展開の推進力は、一つには、従来ばらばらだった部門別業務を相互に結合し、供給業者および顧客とリアルタイムでコミュニケーションしたいという要求に根ざしており、電子商取引と、それに付随するそのような接続性をもたせるためにますます提供されるようになっていく相互接続ソリューションおよび企業間ソリューションへのアクセスのための媒体としてのインターネットの急激な発展によって、さらに拍車がかけている。

【0003】この最近の展開に伴って、最近の企業は多くの異なるオペレーティング・プラットフォームを動的にリンクして、シームレスに相互接続されたシステムを構築することが必要になっている。企業は、合併活動から生じる非集中的購買業務、アプリケーション・ベースの要件、異種技術プラットフォームなどの要因のために、異種情報システムによって特徴づけられることが多い。さらに、供給業者、提携先、顧客の間のリアルタイムの企業外接続性を円滑にしたいという要求が、異種環境において接続性を持たせようという強い動機となっている。

【0004】顧客要件の急激な増大に応じて、情報技術の分野ではそのようなニーズに対応して企業データ・センターのための接続性を拡張するデータ処理ソリューションが開発され始めている。

【0005】本明細書の主題に関連する背景情報は例えば、複数区分のワークロード管理について記載されているIBM資料SG24-5326-00「OS/390 Workload Manager Implementation and Exploitation」(ISBM: 0738413070)、およびESA/390命令セット・アーキテクチャについて記載されているIBM資料SA22-7201-06「ESA/390 Principles of Operation」などである。

【0006】まず最初に、動作上の相互依存性があると考えられる様々なアプリケーションの処理サポートを同時に提供する統合システムを供給する必要があるために、区分多重処理システムの市場が拡大している。単一の物理コンピューティング・システム内で複数のオペレーティング・システム・イメージをサポートする機能を備えた

この種の区分システムは、かつてはメインフレーム・コンピュータ(IBM S/390システムなど)の独壇場だったが、ますます広範囲な供給業者から販売されるようになってきている。たとえば、サン・マイクロシステムズ社は、最近、Ultra Enterprise 10000ハイエンド・サーバでシステム区分化の一形態を提供し始めた。これについては、米国特許第5931938号で詳述されている。他の会社も、このタイプのシステムに対する関心を示す方針書を出している。

【0007】業界のこの動向は、企業内の様々な計算ワークロードを1つの(または少数の)物理サーバ・コンピュータに統合する際と、動的に再構成可能なハードウェア環境において試験レベルのコードと実働レベルのコードを同時に実施するための、システム区分化という「システム内システム」の利点を際立たせている。さらに、前掲の相互参照特許出願に記載されているIBM S/390コンピュータ/システムなどの特定の区分多重処理システムでは、(プロセッサ、メモリ、入出力資源を含む)資源を、システム内で実行されるワークロードに割り当てられた区分に応じて、論理区分内および論理区分間で動的に割り振ることができる(IBMおよびS/390は、国際ナショナル・ビジネス・マシーンス・コーポレーションの登録商標である)。ワークロード優先順位に基づく動的資源割り振りを可能にするこの機能は、従来からデータ・センター管理者がワークロードの一時的な急上昇に対応するために自分の予想計算ワークロードに意図的に余分な量の資源を割り当てる結果につながっている、長年の処理能力計画問題に対処するものである。

#### 【0008】

【発明が解決しようとする課題】これらの区分システムは、企業全体の異種システムを組み込むためのデータ・センターの拡張を容易にするが、現在、この種のソリューションは、異種または同種の区分プラットフォームを単一の相互運用区分システムに機能的に統合する単純な機構を提供しない。それどころか、この種の新型サーバは、オペレーティング・システム・イメージを単一の物理ハードウェア・プラットフォーム内に統合することはできるが、サーバの各区分内にあるオペレーティング・システム間の相互運用性の必要に十分に対処していない。この相互運用性の問題は、様々な区分内に異種オペレーティング・システムを有する異種システムではさらにひどくなる。さらに、この種のシステムは一般に、区分間の高帯域幅で低待ち時間の相互接続を可能にするタイプの異種プラットフォーム間の区分間資源共用に対応していなかった。このような相互運用性問題に対処することが重要な理由は、このような問題の解決策を組み込んだシステムがあれば、別個の区分で実行されているプロセス間の通信のためのより堅固な機構が可能になり、それによって、そのようなアプリケーションが別個のオ

ペレーティング・システム上で稼働しているがそれらのアプリケーションは実際には互いに対してローカルであるということを利用するようにすることができるためである。

【0009】いくつかのオペレーティング・システムの「カーネル」の拡張機能によって、区分間メモリ共用を実施するための共用記憶域の使用を容易にすることが可能である。「カーネル」とは、オペレーティング・システム内の中核システム・サービス・コードである。このようにして形成された境界上でネットワーク・メッセージ受渡しプロトコルを実施することができるが、オペレーティング・システムのうちの1つまたは複数のオペレーティング・システムを修正する手段に訴えずに、効率的なプロセス間通信を可能にすることが望ましい場合が多い。また、米国特許第5931938号に記載されている米国サン・マイクロシステムズのUltra Enterprise 10000ハイ・エンド・サーバのようにメモリ領域を共用するために区分の分離を制限することを回避するのが望ましい場合が多い。それと同時に、区分間での情報の受渡しをネットワーク速度ではなくメモリ速度で行うことが望ましい。したがって、アドレスを共用せずに区分メモリ間で記憶内容を移動させる方法が望ましい。

【0010】IBM S/390 Gbit Ethernet (R) (米国特許第5442802号)の入出力アダプタを使用して、1つの区分のカーネル・メモリから他の区分のカーネル・メモリにデータを移動させることができるが、データは第1カーネル・メモリからアダプタ上の待ち行列バッファに移動され、それからアダプタ上の第2の待ち行列バッファに転送された後、第2のカーネル・メモリに転送される。これは、メモリからメモリに転送するのに合計3回のデータ移動があることを意味する。どのようなメッセージ受渡し通信方式でも、データ・アクセスの待ち時間が共用記憶域との間での1回の記憶取出しの待ち時間に近づくように、データ移動操作の数を最小限にすることが望ましい。移動機能には、転送される各データ・ブロックごとに3回のデータ移動操作がある。これらの操作のうちの1つまたは2つをなくす方法が望ましい。

【0011】同様に、IBM S/390並列シスプレックス結合機構機は、区分間メッセージ受渡しを容易にすることができ、そのために使用される。しかし、この場合、第1のカーネル・メモリから結合機構へ、次に結合機構から第2のカーネル・メモリにデータの転送が行われる。これは、望ましい1回の移動ではなく2回のデータ操作を必要とする。

【0012】多くのコンピュータ・システムでは、無許可または不当なアクセスによるコンピュータ上のデータやアプリケーションの不正使用を防止するように、ユーザの識別を妥当性検査することが望ましい。様々なオペ

レーティング・システムおよびアプリケーション・システムが、そのためのユーザ認証およびその他のセキュリティ・サービスを備えている。区分システム、または実際にはどのようなシステムのクラスタまたはネットワークにであっても、アクセスするユーザをアクセス時点または重要な資源の要求や重要なシステム・メンテナンス機能の実行などの重要なチェックポイントで、一度妥当性検査することが望ましい。この要望を、「シングル・サイン・オン」要件と呼ぶ。このため、様々な区分のセキュリティ・サービスが相互作用するかまたは統合されなければならない。この例としては、ウェブから受け取った「デジタル認証」を処理し、それらをOS/390内の従来のユーザIDおよびパスワード妥当性検査と権限授与にマッピングするOS/390 SAF (RACF) インタフェースの拡張機能、ケルベロス・セキュリティ・サーバ、ディレクトリ・サービスのための最新のLDAP標準などがある。

【0013】さらに、電子商取引の競争的性質のため、ユーザ認証と権限授与は従来のシステムよりもさらに重要である。従業員であれば一日の始めに認証されるのを待つのを覚悟しているであろうが、顧客は認証にあまり時間がかかるとそのまま他の所に行ってしまうかもしれない。暗号の使用は、ウェブの公共的性質のために、この問題をさらに悪化させる。また、1つのオペレーティング・システム内に、他のオペレーティング・システムのためには作成されていないデバイス・ドライバが存在する場合も多い。そのような場合、1つの区分内のデバイス・ドライバに他の区分から効率的な方法でインタフェースすることが望ましい。現在、このタイプの動作のために利用可能な方法はネットワーク接続しかない。

【0014】分散システムの問題の1つは、他のシステムが過剰に利用されている一方で、1つのシステム内の「ホワイト・スペース」または十分に利用されていない資源の管理である。システム間またはシステム・イメージ間で作業を移動する、IBMのLoadLeveler、またはOS/390オペレーティング・システム・ワークロード・マネージャの並列シスプレックス機能などのワークロード・バランス機能がある。区分コンピューティング・システムでは、区分間で作業ではなく資源をシフトさせることが可能であり、望ましい。これが望ましい理由は、機能シフトに伴う大規模な環境切り換えとデータ移動が回避されるからである。

【0015】シスプレックスの外部クラスタ化接続を使用してUNIX (R) オペレーティング・システムのソケット間接続を実現するIBM S/390のための「シスプレックス・ソケット」は、従来技術の一例である。この場合、サービスが、利用可能なセキュリティ・レベルを示し、必要なセキュリティ・レベルを示すアプリケーションの標識に基づいて接続を設定する。しかし、この場合、高いセキュリティ・レベルには暗号化が

設けられ、シスプレックス接続自体が、本発明によって実現されるメモリ接続よりもはるかに深い物理トランスポート層を有する。

【0016】同様に、SSL認証機能を備え、ウェブ・アプリケーション・サーバに（プロキシとして）認証情報を提供するウェブ・サーバは、本発明のメモリ共用または直接メモリ間メッセージを使用すれば有利なもう1つの例と見なすことができる。この場合、そのプロキシは、セキュリティ・サーバに渡すデータを再暗号化する必要がなく、さらに、管理すべき深い接続インタフェースもない。実際に、本発明のこの実施形態では、プロキシ・サーバは本質的に、セキュリティ・サーバと同じオペレーティング・システムの下で稼働するプロキシ・サーバと本質的に同じであるプロセスを介して、セキュリティ・サーバと通信することが、当業者ならわかるであろう。

【0017】

【課題を解決するための手段】従来の技術の上述の問題および欠点は、本発明において克服されるとともに、本発明はその他の有利な特徴を備える。本発明の一態様は複数の異種オペレーティング・システム・イメージをサポートすることが可能な区分コンピュータ・システムであって、それらのオペレーティング・システム・イメージが、記憶場所を共用せずにメモリ内にあるそれらの記憶場所間でメッセージをメモリ速度で並列して受渡しすることができるコンピュータ・システムを含む。これは、1つの区分の1つのカーネル・メモリ空間から第2の区分のカーネル・メモリ空間へのデータの直接移動を容易にする、特別なデバイス・ドライバを備えた入出力アダプタを使用することによって行われる。

【0018】開示の区分セキュリティ・システムは、共通セキュリティ・サーバを含む第1の区分と、セキュリティ・クライアントを有する第2の区分とを有する。区分処理システムはさらに、第1の区分がアクセス可能な第1の部分と、第2の区分がアクセス可能な第2の部分とを有する主記憶域を有する。また、セキュリティ・クライアントに接続され、ユーザがセキュリティ・クライアントに許可要求を送信するための機構も含まれる。セキュリティ・クライアント内の第1の送信部が、前記主記憶域を介してセキュリティ・クライアントから共通セキュリティ・サーバにこの許可要求を送る。共通セキュリティ・サーバ内の第2の送信部が、許可要求に対する応答を前記主記憶域を介して共通セキュリティ・サーバからセキュリティ・クライアントに送る。次に、セキュリティ・クライアント内の第3の送信部が、この応答をセキュリティ・クライアントからユーザに送る。

【0019】本発明の一実施形態では、共用メモリ資源は複数の区分内で実行される複数の相互運用プロセスのために、指定メモリ資源に独立してマップされる。このようにして、共通共用メモリ空間がメモリ資源を共用す

る各区分内のプロセスによってマップされ、区分内でそのプロセスに割り当てられ、通常のプロセス実行の課程でデータの読取りおよび書込みに使用可能なメモリ資源であるかのように見える。

【0020】他の実施形態では、これらのプロセスは相互に依存し、共用メモリ資源は、いずれかのプロセスまたは両方のプロセスが後でアクセスするように、いずれかのプロセスまたは両方のプロセスから記憶することができる。

10 【0021】本発明の他の実施形態では、システムは、各区分内の様々なプロセスを共用メモリ空間に接続するプロトコルを含む。

【0022】本発明の他の実施形態では、1つの区分のカーネル空間から他の区分のカーネル空間へのデータの直接移動は、入出力アダプタによって可能になる。この入出力アダプタは、区分化に関わらず、すべての物理メモリに物理的にアクセスすることができる。入出力アダプタがすべてのメモリにアクセスすることができるのは、区分間での入出力資源共用を可能にする区分コンピュータ・システムにおける機能の自然の結果である。このような共用については、米国特許第5414851号に記載されている。しかし、この新規で発明的なアダプタは、データ移動機構を使用して1つの区分のメモリから別の区分のメモリに直接データを移動する機能を有する。

【0023】本発明の他の実施形態では、カーネル・メモリ間でのデータの移動の機能は、ネットワーク通信アダプタのハードウェアとデバイス・ドライバ内で実施される。

30 【0024】本発明の他の実施形態では、ネットワーク・アダプタは、ローカルであるがメモリ間インタフェースを介した異種セキュア接続のために最適化された、TCP/IPスタックによって駆動される。

【0025】本発明の他の実施形態では、データ移動機構自体は、区分処理システムの通信ファブリックで実施され、さらに直接的なメモリ間転送を容易にする入出力アダプタによって制御される。

40 【0026】本発明の他の実施形態では、データ移動機構は、特権CISC命令のマイクロコードによって制御される。このマイクロコードは、オペランドとして与えられたネットワーク・アドレスおよびオフセットを物理アドレスに変換することができ、それによって、2つの区分内の実アドレスと仮想アドレスとを有する物理アドレス間の移動文字長命令に相当する機能を実行する（IBM S/390 MVC L命令。IBM資料SA22-7201-06「ESA/390 Principles of Operation」参照）。

50 【0027】本発明の他の実施形態では、データ移動機構は、ハイババイザで実行されるルーチンによって制御される。このハイババイザは、全物理メモリへの仮想メ



モリ・アクセスおよび実メモリ・アクセスを行うことができ、オペランドとして与えられたネットワーク・アドレスおよびオフセットを物理アドレスに変換することができ、それによって、2つの区分間の実アドレスと仮想アドレスとを有するアドレス間の移動文字長命令に相当する機能を実行する (IBM S/390 MVC L)。

【0028】複数の区分のうちの1つの区分内のサーバ・プロセスとその他の区分内のクライアント・プロセスの実施により、区分システムは異種単一システム・クライアント・サーバ・ネットワークを実現することができる。既存のクライアント/サーバ・プロセスは一般に、ネットワーク・プロトコル接続によって相互動作するため、本発明のメッセージ受渡し実施形態で容易に実施され、インタフェース変更をせずにパフォーマンスおよびセキュリティ上の利点が得られる。しかし、本発明の共用メモリ実施形態でクライアント/サーバ・プロセスを実施すれば、配備のパフォーマンスまたは速度、あるいはその両方の点で有利になり得る。

【0029】本発明の他の実施形態では、共用メモリまたはメモリ間メッセージ受渡しを使用するアプリケーション・サーバのために信用/保護サーバ環境が提供される。これによって、従来の技術のように追加の暗号化または認証を必要とせずに、許可および認証データの外面化というセキュリティ露出を回避することができる。

【0030】本発明の特定の実施形態では、ウェブ・サーバは、OS/390、Z/OS、またはVM/390の下で稼働する「SAF」セキュリティ・インタフェースへのメモリ・インタフェースを介して通信する、OS/390用Linux Apacheである。この実施形態では、Linuxの「プラグブル・オーセンティケーション・モジュール (PAM: Pluggable Authentication Module)」を修正して、メモリ接続を介してSAFインタフェースを駆動する。

【0031】本発明の他の実施形態では、セキュリティ証明/コンテキストが共用メモリに記憶されるか、またはメモリ間転送によって複製されるように、ポリシー・ディレクタやRACFのようなセキュリティ・サーバに修正を加える。

【0032】

【発明の実施の形態】本発明を構成するとみなされる主題は、本明細書の特許請求の範囲で具体的に示し、明確に請求する。本発明の上記およびその他の目的、特徴、および利点は、以下の詳細な説明を添付図面を参照しながら読めば明らかになる。

【0033】本発明の好ましい実施形態の特定の態様について説明する前に、区分処理システムの基本構成要素について説明すれば有益であろう。この説明を背景として用いれば、本発明の具体的な有利な特徴を区分システムでどのように使用して区分システムのパフォーマンス

を向上させることができるかが、よりよく理解できよう。IBM資料SC28-1855-06「OS/390 V2R7.0 OSA/SF User's Guide」を参照されたい。この資料には、OS/390オペレーティング・システムの一要素であるオープン・システムズ・アダプタ・サポート機構 (OSA/SF) の使用方法が記載されている。この資料は、OSA/SFのセットアップと、OS/2インタフェースまたはOSA/SFコマンドのいずれかを使用してOSAをカスタマイズし、管理するための解説書である。G321-5640-00「S/390 cluster technology: Parallel Sysplex」には、汎用大規模市場向けに開発されたクラスタ化マルチプロセッサ・システムについて記載されている。S/390並列シスプレックス・システムは、スケーラビリティの高いクラスタ化コンピュータ環境における完全データ共用および並列処理の利点を組み合わせるように設計されたアーキテクチャに基づく。並列シスプレックス・システムにより、コスト、パフォーマンス範囲、および可用性の分野において大きな利点が得られる。IBM資料SC34-5349-01「MQSeries Queue Manager Clusters」には、MQSeries待ち行列管理プログラム/クラスタについて記載されており、クラスタの概念、用語、および利点について説明している。この資料には、新しいコマンドおよび変更されたコマンドの構文の概要が記載され、待ち行列管理プログラムのクラスタのセットアップおよびメンテナンスのための作業の例がいくつか示されている。IBM資料SA22-7201-06「ESA/390 Principles of Operation」には、ESA/390アーキテクチャの詳細な定義がリファレンスのために記載されている。この資料は、主としてアセンブラ言語プログラマ向けの解説書であり、各機能について、その機能に依存するアセンブラ言語プログラムを作成するのに必要な詳細度で記載されているが、ESA/390の機能の詳細に関心のある人には有効であろう。

【0034】上記の各資料には従来技術の例が示されており、本発明の背景を理解する上で有用であろう。

【0035】図1を参照すると、区分処理システム100を構成する基本要素が図示されている。システム100は、ブロックAおよびBとして図示されているブロックに区分化することができる物理メモリ資源から成るメモリ資源ブロック101と、区分メモリ資源101と一致するように論理的または物理的に区分化することができる1つまたは複数のプロセッサで構成することができるプロセッサ資源ブロック102と、同様に区分化することができる入出力 (I/O) 資源ブロック103とから成る。これらの区分資源ブロックは、スイッチング・マトリックスなどを含むことができる相互接続ファブリック104を介して相互接続されている。相互接続ファブリック104は、プロセッサ102Bをメモリ101Bに接続するなどの区分内の資源を相互接続する機能を

果たすことができ、プロセッサ 102A をメモリ 101B に接続するなど区分間の資源を相互接続する役割も果たすことができることを理解されたい。本明細書で使用する「ファブリック」という用語は、当技術分野で周知のシステムの各要素を相互接続する方法一般を意味することを意図したものである。これは単純なポイントツーポイント・バスであってもよく、高度なルーティング機構であってもよい。本発明の図面には 2 つの区分 (A および B) を有するシステムを図示するが、このような図は本明細書の説明を簡単にするために選定したものであることと、本発明は、使用可能な資源と同じ数だけ、また区分化技法によって可能な数だけの区分が実現されるように構成可能なシステムを含むものと意図されていることが容易にわかるであろう。

【0036】図の各区分 A および B は別々にして、別個のデータ処理システムの構成要素、すなわち、プロセッサ、メモリ、および入出力資源を構成することが容易にわかるであろう。このことは、区分処理システムに独自の「システム内システム」の利点を与える特徴である。実際に、本明細書で例示するように、現在使用可能な区分処理システム間の主な特徴は、システム資源を区分化することができる境界と、それらの境界を超えて区分間で資源を移動させることができる容易さである。

【0037】区分を分離するが物理的境界である第 1 の事例の最も良い例は、サン・マイクロシステムズの Ultra Enterprise 10000 システムである。Ultra Enterprise 10000 システムでは、区分は物理境界に沿って境界を画定され、具体的には、各システム・ボードがいくつかのプロセッサとメモリと入出力装置とを含む、1 つまたは複数の物理的システム・ボードから成る領域または区分である。領域は、これらのシステム・ボードのうちの 1 つまたは複数のシステム・ボードおよびそれに接続された入出力アダプタであると定義される。各領域は、固有のバスとスイッチのアーキテクチャによって相互接続される。

【0038】図 2 に、物理区分処理システム 200 を構成する要素の高水準図を示す。図 2 を参照すればわかるように、システム 200 は 2 つの領域または区分 A および B を含む。区分 A は、2 枚のシステム・ボード 200A1 および 200A2 から成る。区分 A の各システム・ボードは、メモリ 201A と、プロセッサ 202A と、入出力装置 203A と、相互接続媒体 204A とを含む。相互接続媒体 204A により、システム・ボード 200A1 上の構成要素が互いに通信することができる。区分 B は単一のシステム・ボードから成り、同様の構成処理要素、すなわち、メモリ 201B、プロセッサ 202B、入出力装置 203B、および相互接続媒体 204B を含む。区分にグループ分けされたこれらのシステム・ボードのほかに、各システム・ボードを結合し、1 つの区分内のシステム・ボード間の相互接続と、異なる区

分内のシステム・ボードの相互接続を可能にする相互接続ファブリック 205 がある。

【0039】次のタイプのシステム区分は論理区分と呼ばれるものである。このシステムには、様々な区分への資源の割当を制約する物理境界がなく、システムは、物理的な場所には関係なくいずれの区分にでも割り当てることができる使用可能な資源のプールを有するものと見ることができる。たとえば、所与のシステム・ボード

(システム・ボード 200A1 など) 上のすべてのプロセッサが必ず同じ区分に割り当てられる物理区分システムとの違いである。IBM AS/400 システムが、論理区分専用資源処理システムの例である。AS/400 システムでは、ユーザが所与の区分にプロセッサ、メモリ、および入出力装置をそれらの物理場所に関係なく組み込むことができる。したがって、たとえば物理的に同じカード上にある 2 つのプロセッサを、2 つの異なる区分のための資源として指定することができる。同様に、カードなどの所与の物理パッケージ内のメモリ資源は、そのアドレス空間の一部を論理的に 1 つの区分専用とし、残りの部分を別の区分専用とすることができる。

【0040】AS/400 システムなどの論理区分専用資源システムの特徴は、資源の区分への論理マッピングが、システムの手動再構成によってのみ変更することができる静的に行われる割当てであることである。図 3 を参照すると、プロセッサ 302A1 は、システム内の任意の場所に物理的に配置することができ、論理的に区分 A 専用化されたプロセッサである。ユーザがプロセッサ 302A1 を区分 B にマップし直したい場合、そのプロセッサをオフラインにし、その変更に対応するように手動でマップし直す必要がある。論理区分システムは、たとえば固定数のプロセッサをサポートするシステム・ボードなどの物理的区分化境界という制限による制約を受けないため、資源区分化の細分性が高い。しかし、このような論理区分専用資源システムの再構成は、区分再マッピングの対象となる資源の動作を中断させなければ行うことができない。したがって、このようなシステムは、物理区分システムに固有の制限のいくつかを回避するが、区分間での資源の静的マッピングに付随する再構成上の制約があることがわかる。

【0041】このため、発明人等は、論理区分共用資源システムを考える。このようなシステムの一例は、IBM S/390 コンピュータ/システムである。論理区分共用資源システムの特徴は、プロセッサなどの論理区分資源を複数の区分が共用することができることである。この特徴により、論理区分専用資源システムの再構成上の制約が事実上克服される。

【0042】図 4 に、論理区分資源共用システム 400 の全体構成を示す。論理区分専用資源システム 300 と同様、システム 400 は、システム内の物理的場所に関係なくいずれの区分 (この例では A または B) にでも論

理的に割り当てることができるメモリ 4 0 1 と、プロセッサ 4 0 2 と、入出力資源 4 0 3 とを含む。しかし、システム 4 0 0 でわかるように、特定のプロセッサ 4 0 2 または入出力資源 4 0 3 の論理区分割当ては、「ハイバパイザ」( 4 0 8 ) で稼働しているスケジューラに従って仮想プロセッサ ( 4 0 6 ) および入出力ドライバ ( 4 0 7 ) をスワップすることによって動的に変更することができる。(ハイバパイザとは、仮想機械のために資源のスケジューリングと割振りを行う監視プログラムである。) プロセッサと入出力資源の仮想化により、区分をそれらの資源間で動的に共用することができるようにする適切な優先順位づけを使用して、全オペレーティング・システム・イメージを動作停止中にスワップすることができる。

【 0 0 4 3 】 論理区分共用資源システム 4 0 0 は、プロセッサおよび入出力資源を共用するための機構を備えるが、既存のシステムは区分間メッセージ受渡しに完全には対処していなかった。これは、既存の区分システムは区分間の通信を可能にすることができないということではない。実際には、本明細書に記載のように各タイプの区分システムではこのような通信が行われる。しかし、これらの実施態様のいずれも、ハイバパイザ、共用メモリ実施態様、または区分間を接続する標準アダプタまたはチャネル通信装置のセットまたはネットワークの介入なしにカーネル・メモリからカーネル・メモリにデータを移動する手段を備えていない。

【 0 0 4 4 】 米国特許第 5 9 3 1 9 3 8 号に記載されているようなサン・マイクロシステムズの Ultra Enterprise 10000 に代表される物理区分多重処理システムでは、マスク／レジスタを適切に設定することによって、システム・メモリの一領域にハードウェア・レベルで複数の区分がアクセスすることができる。サンの特許では、区分間ネットワークのためのバッファ機構および通信手段として使用できると記載している以外には、この機能の利用法を教示していない。

【 0 0 4 5 】 「Coupling Facility Configuration Options: A Positioning Paper」( G F 2 2 - 5 0 4 2 - 0 0 , I B M コーポレーション ) で詳述されているように、I B M S / 3 9 0 システムでは、共通にアドレス指定された物理メモリを「統合結合機構」として使用する同様の内部クラスタ化機能について記載されている。この場合、共用記憶域は実際にリポジトリであるが、この共用記憶域への接続は X C F と呼ぶデバイス・ドライバのような入出力資源を介する。この場合、共用メモリは結合機構で実現されるが、非 S / 3 9 0 オペレーティング・システムはそれを使用する拡張部を作成する必要がある。さらに、この実施態様では、データを 1 つの区分のカーネル・メモリから結合機構のメモリに移動させ、次に第 2 の区分のカーネル・メモリに移動させる。

【 0 0 4 6 】 カーネルとは、オペレーティング・システ

ムのうち、ハードウェア資源の割振りなどの基本機能を実行する部分である。カーネル・メモリとは、カーネルがその機能を実行するために使用する、カーネルが使用可能なメモリ空間である。

【 0 0 4 7 】 それに対して、本発明は、区分またはハードウェア内にオペレーティング・システムの共用記憶域拡張部を設けずに、新しい入出力アダプタおよびそのデバイス・ドライバを使用可能にする機能を使用して 1 回の動作で 1 つの区分のカーネル・メモリから別の区分のカーネル・メモリにデータを移動する手段を設ける。

【 0 0 4 8 】 本発明がどのように実現されるかを理解するには、オペレーティング・システムにおけるプロセス間通信について理解すれば役に立つ。図 5 を参照すると、プロセス A ( 5 0 1 ) および B ( 5 0 3 ) がそれぞれ、アドレス空間であるメモリ A ( 5 0 2 ) およびメモリ B ( 5 0 4 ) を有する。これらのアドレス空間は、カーネル ( 5 0 5 ) によるシステム呼出の実行によってそれらに割り振られる実メモリを有する。カーネルは、それ自体のアドレス空間であるメモリ K ( 5 0 6 ) を有する。通信の一形態では、プロセス A と B は、バッファ 5 1 0 の作成、接続、アクセスを行う適切なシステム呼出しを行うことにより、メモリ K 内にバッファ 5 1 0 を作成することによって通信する。これらの呼出のセマンティクスはシステムによって異なるが、その効果は同じである。第 2 の通信形態では、メモリ S ( 5 0 7 ) のセグメント 5 1 1 を、メモリ A ( 5 0 2 ) およびメモリ B ( 5 0 4 ) のアドレス空間にマップする。このマッピングが完了した後は、プロセス A ( 5 0 1 ) および B ( 5 0 3 ) は、両方のプロセスが認識する任意のプロトコルに従ってメモリ S ( 5 0 7 ) の共用セグメントを自由に使用することができる。

【 0 0 4 9 】 図 6 で、プロセス A ( 6 0 1 ) とプロセス B ( 6 0 3 ) とが異なるオペレーティング・システム・ドメイン、イメージ、または区分 ( 区分 1 ( 6 1 4 ) および区分 2 ( 6 1 5 ) ) にある。ここでは、カーネル・メモリとしてメモリ K 1 ( 6 0 6 ) およびメモリ K 2 ( 6 0 8 ) を有するカーネル 1 ( 6 0 5 ) およびカーネル 2 ( 6 0 7 ) がある。この場合、メモリ S ( 6 0 9 ) は、区分 1 と区分 2 の両方がアクセスすることができる物理メモリの空間である。このような共用は、U E 1 0 0 0 0 メモリ・マッピング実施態様または S / 3 9 0 ハイバパイザ実施態様などを限定せずに任意の実施態様に従って、または区分化によって作成されるアクセス障壁を限定するその他の手段に従って、使用可能化することができる。他の例として、共用空間を定義する構成レジスタ内に先頭アドレスのある、最上位物理メモリ・アドレスに共用メモリをマップする。

【 0 0 5 0 】 規則として、メモリ S ( 6 0 9 ) は、メモリ K 1 およびメモリ K 2 にマップされるカーネル 1 およびカーネル 2 の拡張部によって使用される共用セグメン

ト ( 6 1 0 ) を有する。セグメント 6 1 0 を使用して、メモリ K 1 ( 6 0 6 ) およびメモリ K 2 ( 6 0 8 ) にマップされて、前述の第 1 の形態に従って区分間通信を可能にする、メモリ ( 6 0 9 ) のセグメントのための定義および割振りテーブルを保持するか、または、図 5 を参照しながら前述した第 2 の通信形態に従ってメモリ A ( 6 0 2 ) およびメモリ B ( 6 0 4 ) にマップされるセグメント S 2 ( 6 1 1 ) を定義する。本発明の一実施形態では、メモリ S はサイズが限定されており、実記憶域に固定される。しかし、付随するページ管理タスクが効率的に管理される限り、メモリは固定される必要はなく、それによってより大きな共用記憶空間が可能になるものと企図される。

【 0 0 5 1 】 本発明の第 1 の実施形態では、共用記憶域の定義および割振りテーブルは、共用メモリ構成データ・セット ( S M C D S ) ( 6 1 3 ) からデータを読み取り、メモリ S ( 6 0 9 ) のセグメント S ( 6 1 0 ) 内にテーブルを作成する共用メモリ構成プログラム ( S M C P ) ( 6 1 2 ) と呼ばれるスタンドアロン・ユーティリティ・プログラムによってメモリ内に設定される。したがって、どのカーネルが記憶域のどのセグメントを共用するかの割振りと定義は固定されており、このユーティリティによって作成された構成によって事前決定される。その後、様々なカーネル拡張部がその共用記憶域を使用して、パイプ、メッセージ待ち行列、ソケット、さらには、一部のセグメントをそれ独自の規則に従って共用メモリ・セグメントとしてユーザ・プロセスに割り振ることなど、様々なイメージ間、プロセス間通信構成物を実現する。このようなプロセス間通信は、I P C A P I 6 1 8 および 6 1 9 を介して使用可能にされる。

【 0 0 5 2 】 共用記憶域の割振りテーブルには、イメージ識別子、セグメント番号、グループ I D、ユーザ I D、「スティッキ・ビット」、および許可ビットから成る項目が含まれる。スティッキ・ビットは、関連する記憶域がページ可能でないことを示す。この例示の実施形態では、スティッキ・ビットは予約済みであり、1 をとる (すなわち、データはメモリ内のその場所に固定すなわち「スティック」される)。セグメントを使用する各グループ、ユーザ、およびイメージは、このテーブル内に項目を持つ。規則として、すべてのカーネルがテーブルを読み取ることができるが、どのカーネルもテーブルに書き込むことはできない。初期設定時に、カーネル拡張部は構成テーブルを読み取り、他のプロセスによってイメージ間プロセス間通信が要求されたときに使用するためにそれ独自の割振りテーブルを作成する。カーネルは、割り振られた空間の一部または全部を、プロセス間通信を要求する他のプロセスの要求に作成する

「パイプ」、ファイル、およびメッセージ待ち行列の実現のために使用する。パイプは、1 つのプロセスからカーネル機能を介して第 2 のプロセスに向けて送られるデ

ータである。パイプ、ファイル、イメージ待ち行列は、Linux、OS/390 USS、およびほとんどの UNIX ( R ) オペレーティング・システムで使用されているような、標準 UNIX ( R ) オペレーティング・システムのプロセス間通信 A P I およびデータ構造である。共用空間の一部は、直接クロス・システム・メモリ共用のために、さらに他のカーネル拡張部によって他のプロセスのアドレス空間にマップすることができる。

【 0 0 5 3 】 共用メモリの割振り、使用、および仮想アドレス空間へのマッピングは、各カーネルがそれ自体の規則と変換プロセスとに従って行うが、基本的なハードウェア・ロックとメモリ共用プロトコルは、システムの他の部分の基礎にある共通ハードウェア設計アーキテクチャによって駆動される。

【 0 0 5 4 】 より高水準のプロトコルは、通信が行われるためには共通でなければならない。好ましい実施形態では、これは、UNIX ( R ) オペレーティング・システムと共に使用するために、要求をクロス・イメージであると識別する拡張部分を付けて、様々なオペレーティング・システム・イメージのそれぞれに I P C ( プロセス間通信 ) A P I を実施させることによって行われる。この拡張部分は、パラメータによって、または別個の新しい識別子/コマンド名によって行うことができる。

【 0 0 5 5 】 図 4 および図 7 を参照すると、本発明は、チャネルまたはネットワーク接続を介したデータの転送と、オペレーティング・システムの共用メモリ拡張部の使用の両方を回避することがわかる。区分 7 1 4 内のアプリケーション・プロセス ( 7 0 1 ) が、ソケット・インタフェース 7 0 8 にアクセスし、ソケット・インタフェース 7 0 8 がカーネル 1 ( 7 0 5 ) を呼び出す。ソケット・インタフェースは、TCP/IP スタックの特定のポートをリスト・ユーザ・プロセスに関係づける構造体である。カーネルは、デバイス・ドライバ ( 7 1 6 ) にアクセスし、デバイス・ドライバ 7 1 6 は、入出力アダプタ ( 7 2 0 ) のハードウェアを介してカーネル・メモリ 1 ( 7 0 6 ) からカーネル・メモリ 2 ( 7 0 8 ) に、メモリ ( 4 0 1 ) にとってメモリ間移動のように見えるようにデータを転送し、区分 7 1 4 および 7 1 5 のプロセッサ ( 4 0 2 ) またはファブリック ( 4 0 4 ) あるいはその両方で実現されたキャッシュ・メモリを迂回する。データを移動した後、入出力アダプタは区分 7 1 5 内のデバイス・ドライバ ( 7 1 7 ) にアクセスし、データが移動されたことを示す。デバイス・ドライバ 7 1 7 は、次に、カーネル 2 ( 7 0 7 ) に、ソケット ( 7 1 9 ) にそのソケットを待っているデータがあることを示す。次に、ソケット ( 7 1 9 ) がそのデータをプロセス ( 7 0 3 ) に示す。したがって、直接メモリ間移動が行われると同時に、外部インタフェース上のデータの移動が回避され、メモリ共用のためのいずれのオペレーティング・システムの拡張も回避される。

【0056】それに対して、図8に示す従来技術のシステムは、別々のメモリ移動動作を使用してカーネル・メモリ1(706)からアダプタ・メモリ・バッファ1(721)に移動する。第2のメモリ移動動作によって、データをアダプタ・メモリ・バッファ1(721)からアダプタ・メモリ・バッファ2(722)に移動する。次に、第3のメモリ移動動作によって、アダプタ・メモリ・バッファ2(722)からカーネル・メモリ2(708)にデータを移動する。これは、3回の別個のメモリ移動動作を使用して2つのカーネル・メモリ間でデータを移動することを意味する。それに対して、図7の本発明では、1回のメモリ移動動作で、データがカーネル・メモリ1(706)とカーネル・メモリ2(708)の間で直接移動する。これは、ユーザ・プロセスから見たときに待ち時間を短縮する効果がある。

【0057】本発明の他の実施形態を図4および図9に示す。ここでは、ファブリック(404)で実データ移動機構ハードウェアが実現される。この実施形態の動作は、上記の説明のように進むが、入出力アダプタ820内の制御状態(822)に従って、データが実際にはファブリック(404)内の移動機構ハードウェアによって移動される点異なる。

【0058】このようなファブリックに配置されたデータ移動機構の一例は、米国特許第5269009号に記載されている。この参照特許に記載されている機構は、区分の主記憶域場所間でのデータ転送を含むように拡張されている。

【0059】実施形態を問わず、本発明は以下の要素を含む。すなわち、CPUの設計によって規定された基礎共通データ移動プロトコルと、入出力アダプタまたはファブリックあるいはその両方のハードウェアと、入出力アダプタへのインタフェースを実現する異種セット・デバイス・ドライバと、本実施形態ではソケット・インタフェースとして示されている共通高水準ネットワーク・プロトコルと、ネットワーク・アドレスを入出力アダプタ(820)が各区分のカーネル・メモリおよびデバイス・ドライバと通信するために使用する物理メモリ・アドレスおよび入出力割込みベクトルまたはポインタにマッピングすることとである。

【0060】データ移動機構は、ハードウェア状態機械として入出力アダプタ内で実現するか、またはマイクロコードとマイクロプロセッサとで実現することができる。あるいは、入出力アダプタによって制御される、機

械の通信ファブリック内のデータ移動機構を使用するなどで実現することができる。このようなデータ移動機構の一例は、米国特許第5269009号に記載されている。

【0061】図10を参照すると、実施態様を問わず、データ移動機構は以下の要素を有する。メモリからのデータがソース・レジスタ(901)に保持され、そのデータがデータ・アライナ(902および904)を介して宛先レジスタ(903)に渡され次にメモリへ返される。したがって、メモリ・フェッチがあり、その後でメモリ・ストアが、連続動作の一部としてある。すなわち、メモリ・ラインから複数のワードがフェッチされると、位置合わせプロセスが行われる。位置合わせされたデータは、メモリ・ストアが開始されるまで宛先レジスタ(903)にバッファリングされる。ソース・レジスタ(901)と宛先レジスタ(903)を使用して、移動動作中にフェッチとストアの間でどの程度の重なり合いが可能かに応じてメモリ・データの単一のラインまたは複数のラインを保持することができる。メモリのアドレス指定は、移動中にフェッチ・アドレスとストア・アドレスを追跡するカウンタ(905および906)によって行われる。制御およびバイト・カウント要素(908)が、アライナ(902および904)を介したデータの流れを制御し、メモリ・アドレスに対するソース・カウンタ(905)または宛先カウンタ(906)の選択(907)が行われるようにする。このコントローラ(908)は、アドレス・カウンタ(905および906)の更新も制御する。

【0062】図11を参照すると、データ移動機構は、デバイス・ドライバによって実施される特権CISC命令(1000)として実現することもできる。このようなCISC命令は、S/390ページ移動、文字長移動などの区分内データ移動にハードウェア機能を利用するが、テーブル・マッピング・ネットワーク・アドレスと、物理メモリ・アドレスまでのオフセットに従って、物理的にメモリをアドレス指定する特権も有することになる。最後に、データ移動機構およびアダプタは、仮想アダプタとして機能するハイババイザ・コードによって実現することができる。

【0063】図12に、データ移動機構がアダプタ内にある場合の、データ移動機構の以下のステップから成る動作を示す。

1101 ユーザが、以下を供給するデバイス・ドライバを呼び出す。

ソース・ネットワークID

ソース・オフセット

宛先ネットワークID

1102 デバイス・ドライバがアドレスをアダプタに転送する

1103 アダプタがアドレスを変換する

IDから物理基底アドレスを探索する(テーブル・ルックアップ)

- ロックおよび現行宛先オフセットを入手する  
 オフセットを加える  
 境界を調べる  
 1104 アダプタがカウントとアドレスをレジスタにロードする  
 1105 アダプタがデータ移動を実行する  
 1106 アダプタがロックを解放する  
 1107 アダプタがデバイス・ドライバに通知し、デバイス・ドライバがユ  
 ーザに「戻る」

きるプロセッサ通信ファブリック内で実施されるデータ

【0064】図13に、以下の方法を使用することがで 10 移動方法を示す。

- 1201 ユーザが、以下を供給するデバイス・ドライバを呼び出す。  
 ソース・ネットワークID  
 ソース・オフセット  
 宛先ネットワークID  
 1202 デバイス・ドライバがアダプタにアドレスを送る  
 1203 アダプタがアドレスを変換する  
 IDから物理基底アドレスを探索する（テーブル・ルックアップ）  
 ロックおよび現行宛先オフセットを入手する  
 オフセットを加える  
 境界を調べる  
 アダプタがロックと物理アドレスをデバイス・ドライバに返す  
 1204 デバイス・ドライバがデータ移動を実行する  
 1205 デバイス・ドライバがロックを解放する  
 1206 デバイス・ドライバが戻る

【0065】以上、区分コンピュータ・システムにおい  
 て異種相互動作を実現する2通りの方法について述べ  
 た。一方は、共用メモリ機構とオペレーティング・シス  
 テム・カーネルの拡張部とを使用して、区分間プロセス  
 間通信プロトコルを使用可能にし、他方は共用入出力ア  
 ダプタの機能を仕様して、全物理メモリをアドレス指定 30  
 し、1回の動作でメモリ間メッセージ受渡しを実現す  
 る。

【0066】以上の構成は、単一システム・クライアン  
 ト／サーバ・モデルを利用したいいくつかの新規な実施態  
 様を生み出す。この構成を実現する1つの方法は、サー  
 バ作業待ち行列を共用記憶空間に入れ、様々なクライア  
 ントが要求を付加することができるようにすることであ  
 る。その場合、「リモート」クライアントのための戻り  
 バッファも共用メモリ空間に入れ、それによってクライ  
 アントがその中に入っている情報にアクセスすることが 40  
 できるようにしなければならない。あるいは、前述のメ  
 ッッセージ受渡し方式を使用して、既存のネットワーク指  
 向クライアント／サーバを迅速かつ容易に配備すること  
 ができる。これらの実施態様は、例示として示したもの  
 であり、新規かつ発明的であるが、限定的なものとな  
 してはならない。実際に、当業者なら、この構成を基に  
 して様々な方法で単一のシステム・パラダイム内で異な  
 るタイプの異種クライアント／サーバ・システムを実現  
 することができ、実現するであろうことが容易にわかる  
 であろう。

【0067】区分のクラスタのワークロード管理

図14を参照すると、OS/390オペレーティング・  
 システムのワークロード・マネージャ（WLM）（13  
 08）は、S/390の区分ハイパバイザと通信して、  
 各区分の割り振られた資源を調整することができる。こ  
 れはLPARクラスタ化と呼ばれる。しかし、非OS/  
 390区分（1301）の場合、WLMは、使用率と、  
 ハイパバイザが供給することができるその他の情報だ  
 けに基づいて割り振りを行わなければならない。区分のオペ  
 レーティング・システムまたはアプリケーションには基  
 づかない。上記の低待ち時間区分間通信（1305）を  
 使用して、情報を区分からWLM（1308）にパイプ  
 すれば、WLM（1308）にシステム資源間割り振りの  
 より効果的な作業を行うのに必要な情報を提供するきわ  
 めてオーバーヘッドの少ない手段となる。これは、アプ  
 リケーションがワークロード管理の装備を備えていない  
 場合であっても有効になり得る。その理由は、制御され  
 るシステムは一般に、システムを出入りするIPパケッ  
 トをカウントするTCP/IPスタック内のパケット活  
 動カウンタにアクセスするコマンド、UNIX（R）オ  
 ペレーティング・システムの「NETSTAT」（UN  
 IX（R）オペレーティング・システム標準コマンド・  
 ライブラリの一部）を実施することができ、また、使用  
 中と遊休状態のサイクルをカウントし、使用率データ  
 （1302）を生成するカーネル内のシステム活動カウ  
 ンタにアクセスする標準UNIX（R）オペレーティン 50



グ・システム・コマンド UNIX(R)オペレーティング・システムの「VMSTAT」(UNIX(R)オペレーティング・システム標準コマンド・ライブラリの一部)も実行することができるためである。必ずしも既存のNETSTATコマンドおよびVMSTATコマンドを使用する必要はなく、パケット・カウントと使用率を供給する基礎機構を使用して資源とパス長のコストを最小限にするのが最もよいということがわかるであろう。このデータを「速度」メトリック(1303)に統合し、それをワークロード・マネージャ(WLM)区分(1307)に送ることによって、WLM(1308)はハイパーバイザに資源の調整を行わせることができる。CPU使用率が高く、パケット・トラフィックが低い場合、その区分はより多くの資源を必要とする。接続(1304および1306)は、相互接続(1305)の実施形態によって異なる。共用メモリ実施形態では、これらの接続はUNIX(R)オペレーティング・システムのPIPE、メッセージQ、SHMEM、またはソケット構成とすることができる。データ移動機構実施形態では、これらは一般には、ソケット接続となる。

【0068】本発明の一実施形態では、「速度」メトリックは、以下のようにして得られる(IBMレッドブック資料番号SG24-4810-01「Understanding RS/6000 Performance and sizing」に記載されているUNIX(R)オペレーティング・システム・コマンドNETSTATおよびVMSTATを参照)。

(NETSTAT)合計パケット数のインターバル・データを使用してスループットを求める。インターバルCPUデータ(VMSTAT)を使用してCPU使用率を求める。これらのデータをプロットし、ピークを1として正規化したトラフィックと共に表示する(1401)。

トラフィックとCPUとの累積相関分析を行う(1402)。トラフィックの関数を関数 $T(C)$ に曲線当てはめする。この例(1402)では、 $T(C) = 0.864 + 1.12C$ である。 $S = dT/dC$ が速度メトリックであり、この例では $S = 1.12$ である。 $S$ が傾向線よりも小さい場合、資源が必要とされる。

【0069】図15の例では、これを2回(1403および1404)行う。管理図が、業界で監視プロセスを作成する標準的な方法である。1405の管理図のように $S$ を動的にプロットする。上記でパケット・トラフィックとCPUとの関係について見たような関係を考えると、収集したデータを統計的管理理論に基づいて様々な方法で監視し、配列することができる。これらの方法は一般に、アクションを引き起こす、制御変数のしきい値に依存する。あらゆるフィードバック・システムと同様に、管理不能状態に近いと判断されたらただちにアクションを起こす必要がある。そうしないとシステムが不安定になる可能性がある。本発明では、これは、内部通信による低待ち時間接続によって行われる。

【0070】静的環境では、 $S$ を使用して、より多くの資源が必要になる使用率を設定することができる。これが機能する限り、平均を超える $S$ もワークロードと時間との関数である。図15を参照すると、まず、これは50%と60%との間にあるように見えることがわかり、次に、 $S$ の谷に続いて少なくとも1時間間隔で使用率のピークが現れることがわかる。したがって、 $S$ は資源のより適時な調整を可能にする「先行標識」であるため、WLMは使用率ではなく $S$ を供給した方がより効果的である。区分計算機の資源は区分間で共用されるため、ワークロード・マネージャはこの $S$ データを複数の区分から入手しなければならない。きわめて低いオーバーヘッドおよび高速でデータの転送を行う必要がある。本発明は、この両方の条件を可能にする。図14を参照すると、ワークロード・マネージャのない区分(1301)で、モニタが使用率とパケット・データを監視し(1302)、それがプログラム・ステップで使用され(1303)、パラメータ(この例では $S$ )が評価される。次にプログラムは低待ち時間区分間通信機構(1305)への接続(1304)を使用し、低待ち時間区分間通信機構はワークロード・マネージャがある区分(1307)内の接続(1306)にそれを渡し、これが「論理区分クラス・マネージャ」(1308)に接続して入力データを渡す。論理区分クラス・マネージャについては、米国特許第09/677338号に記載されている。

【0071】この場合、区分データをワークロード・マネージャに伝える最も効率的な方法は、メモリ共用によるものであるが、ソケット待ち時間がデータの送達時間の余裕を持たせることができるほど低い場合には、内部ソケット接続も有効である。これは、ワークロードと、必要な制御細分性の両方に依存する。

【0072】上記は、ワークロード・マネージャが資源を割り振るための情報を供給する新規で発明的な方法であるが、決して限定的なものとはならない。この例は、多くの新しいコードを使用せずに、すべてではないがほとんどのオペレーティング・システムから得られるメトリックであるため選定している。クライアント・システムは、応答時間やユーザ数など、WLMサーバに渡す任意のメトリックの任意の計測を実施することができる。

【0073】間接入出力デバイス・ドライバが、ハードウェアによってサポートされている可能なオペレーティング・システムのうちの1つのオペレーティング・システムのみで使用可能な場合がある。共用メモリにデバイス・ドライバ・メモリ・インタフェースを渡し、すべての接続システムがドライバ・プロトコルを遵守することによって、装置を複数のシステムで共用することができる。実際には、1つの区分が他の区分のためのI/Oになることができる。装置へのアクセスは、装置を過負荷

にすると単一システムから過負荷にするのと同じ悪い結果になるという認識のもとで、単一システム・レベルに近い。図 16 を参照すると、デバイス・ドライバ (1501) は、共用メモリ (1511) を介してアプリケーションおよびアクセス方法 (1503) からの入出力サービス要求に応答する。

【0074】メッセージ受渡し実施態様はある種の装置には使用することができるが、ソケット、スタック、およびデータ移動の待ち時間を受容しなければならないであろう。これは、ネイティブ装置とネットワーク接続装置の間と見ることができる。

【0075】デバイス・ドライバを稼働させているシステム・イメージに割り振られたプロセッサ資源を、アプリケーションを稼働させているシステム・イメージに割り振られたプロセッサ資源から分離すれば、さらなる機能強化が得られる。これを行った場合、入出力割込みの対象ではないプロセッサにおける、入出力割込みおよびそれに付随するコンテキスト切り換えによるキャッシュおよびプログラムの流れの中断が回避される。

【0076】共通セキュリティ・サーバアプリケーションがウェブ対応になり、統合化されるにつれて、ユーザの検証と権限の設定が従来のシステムよりも浸透するようになる。この計算が、異種システムをまとめてアプリケーションを統合するのに必要である。その結果、LDAP、ケルベロス、RACF、およびその他のセキュリティ機能を統合された方式を使用するには、通常、セキュリティ機能を実行するために共通セキュリティ・サーバへのネットワーク接続が必要である。これはパフォーマンスに影響する。また、ネットワーク・スニファのセキュリティ露出もある。共通セキュリティ・サーバを共用メモリ接続またはメモリ移動機構接続を介してウェブ・サーバに接続すれば、この活動を大幅に高速化することができ、接続が内面化され、セキュリティが向上する。さらに、このような環境では、他のUNIX (R) オペレーティング・システムよりも、パスワード保護に基づくS/390「RACF」またはその他のOS/390「SAF」インタフェース・ユーザ認証の強化されたセキュリティの方を選ぶ顧客もいると考えられ、Linuxの場合には特にそうである。Linuxシステムでは、そのような共用サーバのためのクライアント側の構築が比較的容易であるが、それはユーザ認証が、調整とカスタマイズが行われることを意図した「プラグابل・オーセンティケーション・モジュール (Pluggable authentication module)」によって行われるためである。この場合、セキュリティ・サーバには共用メモリ・インタフェースまたはメモリ間データ移動機構インタフェースを介してアクセスし、ウェブ・サーバがこのインタフェースを争奪する。その結果の作業待ち行列は、セキュリティ・サーバが共用メモリ・インタフェースを介して必要に応じて応答することによって稼

働する。その結果、ウェブ・アプリケーションのセキュリティとパフォーマンスが向上する。図 17 を参照すると、セキュリティ・サーバ (1601) は、共用メモリ (1611) を介してユーザ・プロセス (1603) からのアクセス要求に応答する。ユーザ・プロセスは、カーネル 2 (1607) 内のセキュリティ・クライアント・プロセス (これはLinuxの場合のPAMである) への標準インターネット・プロセス通信 (IPC) インタフェースを使用し、次にこれが共用メモリ (1610) を介してカーネル 1 (1605) 内のカーネル・プロセスに連絡し、さらにこのカーネル・プロセスが、ユーザ・プロセス (1603) のプロキシとしてセキュリティ・サーバ・インタフェース (OS/390またはZ/OSの場合のSAF) を駆動し、共用メモリ (1610) を介してカーネル 2 (1607) 内のセキュリティ・クライアントに許可を返す。

【0077】本発明は、データを自由に流したり、追加の暗号化や認証を必要とするものよりもはるかに安全な共用メモリを使用して、アプリケーション・サーバに提供可能な信用/保護環境を向上させる。

【0078】本発明は多くの改良をもたらす。たとえば、SSL認証機能を備え、ウェブ・アプリケーション・サーバ、Linux Apache、従来のアプリケーション (OS/390) に認証情報を提供する (SAFをPAMに結びつける) ウェブ・サーバや、セキュリティ証明/コンテキストを各プラットフォーム上で公開された既存のセキュリティ・マネージャAPIによって本発明の共用メモリに記憶することができるセキュリティ・マネージャ (すなわちポリシー・ディレクタまたはRACF) である。

【0079】本発明の他の実施形態では、共用メモリに入れられたデータを、単一動作データ移動機構を介してカーネル・メモリ 1 (1606) とカーネル・メモリ 2 (1608) の間で移動させ、それによって共用メモリの作成を回避するだけでなく、ネットワーク接続も回避する。

【0080】本発明のセキュリティ・サーバにおいて、第1の区分 (1614) で共通セキュリティ・サーバ (1601) が稼働し、少なくとも1つの第2の区分 (1615) で少なくとも1つのセキュリティ・クライアント (またはプロキシ) (1603) が稼働する区分処理システムのためにセキュリティを設ける通信ステップを実施する例を以下に示す。

【0081】ユーザ (1650) が認証を要求する。ユーザは、この要求を当技術分野で周知の任意の手段によって渡す。ユーザは、たとえば、端末に接続されたキーボード、タッチ・スクリーン技法、または音声変換を使用して、要求を入力することができる。ユーザは、要求を実行の一部とするプログラムで要求を渡すこともできる。セキュリティ・クライアント (1603) が、ユー



ザからパスワードを受け取る。セキュリティ・クライアントは、この要求をセキュリティ・サーバ(1610)がアクセス可能な記憶場所に入れ、それを行ったことを通知する。第1の区分(1614)内の「セキュリティ・デーモン」がこの通知を認識し、第1の区分(1614)で「プロキシ」クライアント(1616)を開始する。プロキシ(1616)クライアントは、セキュリティ・サーバ(1601)固有のインタフェースを使用して、この要求でセキュリティ・サーバを呼び出す。セキュリティ・サーバ(1601)は、要求を処理し、プロキシ・クライアント(1616)にサーバ応答を返す。プロキシ・クライアントは、セキュリティ・サーバの応答を、第2の区分内のセキュリティ・クライアントがアクセス可能なメモリに入れ、それを行ったことを通知する。この通知によって、セキュリティ・クライアント

(1603)が覚醒し、許可を指し示す。セキュリティ・クライアント(1603)は、この応答をユーザに返す。一実施形態では、第2の区分(1615)内のセキュリティ・クライアント(1603)は、共用メモリ・インタフェース(1609)を使用して第1の区分(1614)内のセキュリティ・サーバ(1601)と通信し、ネットワーク接続のセキュリティ露出を回避し、パフォーマンスを向上させる。他の実施形態では、第2の区分内のセキュリティ・クライアントは、図9に示すデータ移動機構(821)を使用した内部メモリ間移動によって、第1の区分内のセキュリティ・サーバと通信する。図9を参照すると、この第2の実施形態は、セキュリティ・クライアントをプロセスA(803)として実施し、セキュリティ・プロキシはプロセスB(801)として実施して、外部のネットワーク接続を回避し、共用メモリの実施も回避する。

【0082】以上、本明細書では、好ましい実施形態を図示して詳述したが、本発明の主旨から逸脱することなく様々な修正、追加、代用などを行うことができ、したがってそれらも特許請求の範囲で規定されている本発明の範囲に入るものとみなされることが、当業者ならわかるであろう。

【0083】まとめとして、本発明の構成に関して以下の事項を開示する。

【0084】(1) 共通セキュリティ・サーバを含む第1の区分とセキュリティ・クライアントを含む第2の区分とを有する区分処理システムにおいてセキュリティを設ける方法であって、

a) ユーザによる許可要求を前記第2の区分内の前記セキュリティ・クライアントに送るステップと、

b) 前記許可要求を前記セキュリティ・クライアントから前記第1の区分内の前記共通セキュリティ・サーバに送るステップと、

c) 前記許可要求に対する第1の応答を前記第1の区分内の前記共通セキュリティ・サーバから前記第2の区分

内の前記セキュリティ・クライアントに送るステップとを有し、区分間の前記要求または前記第1の応答のうちのいずれか一方の前記送信が主記憶域を経由し、

d) さらに、前記セキュリティ・クライアントから前記ユーザに第2の応答を送るステップとを含む方法。

(2) ステップbの許可要求の送信が、

b1) 前記第2の区分内で稼働する前記セキュリティ・クライアントが信号を送って前記第1の区分で稼働している第1のプログラムに前記第1の区分内のプロキシ・クライアントを始動させるステップと、

b2) 前記プロキシ・クライアントから前記第1の区分内の前記セキュリティ・サーバに前記要求を送るステップとをさらに含む、上記(1)に記載の区分処理システムにおいてセキュリティを設ける方法。

(3) ステップbまたはステップcのいずれかが、前記第1の区分と前記第2の区分との間で共用される主記憶域を使用するステップを含む、上記(1)に記載の区分処理システムにおいてセキュリティを設ける方法。

(4) ステップbまたはステップcのいずれかが、メモリ間データ移動によって前記第1の区分と前記第2の区分との間にリンクされた主記憶域を使用するステップを含む、上記(1)に記載の区分処理システムにおいてセキュリティを設ける方法。

(5) ステップbが、前記セキュリティ・サーバ固有のインタフェースを使用した前記プロキシ・クライアントによるプログラム呼出しを含む、上記(1)に記載の区分処理システムにおいてセキュリティを設ける方法。

(6) ステップcが、前記共通セキュリティ・サーバから前記セキュリティ・クライアントに前記第1の区分内で稼働している第1のプログラムを介して前記第1の応答を送るステップをさらに含む、上記(1)に記載の区分処理システムにおいてセキュリティを設ける方法。

(7) 前記第2の区分内の前記セキュリティ・クライアントから前記ユーザに送られる前記第2の応答がプログラム処置である、上記(1)に記載の区分処理においてセキュリティを設ける方法。

(8) 共通セキュリティ・サーバを含む第1の区分とセキュリティ・クライアントを有する第2の区分とを有するセキュリティを設ける区分処理システムであって、ユーザによる許可要求を前記第2の区分内の前記セキュリティ・クライアントに送る手段と、前記許可要求を前記セキュリティ・クライアントから前記第1の区分内の前記共通セキュリティ・サーバに送る手段と、前記許可要求に対する第1の応答を前記第1の区分内の前記共通セキュリティ・サーバから前記第2の区分内の前記セキュリティ・クライアントに送る手段とを有し、区分間の前記要求または前記第1の応答のうちのいずれか一方の前記送信が主記憶域を経由し、さらに前記セキュリティ・クライアントから前記ユーザに第2の応答を送る手段とを含む区分処理システム。

( 9 ) 前記許可要求を送る前記手段が、前記第 1 の区分内で稼働し、プロキシ・クライアントを始動させる第 1 のプログラムと、前記第 2 の区分内で稼働している前記セキュリティ・クライアントによって前記第 1 のプログラムに信号を送り、それによって前記第 1 の区分内の前記プロキシ・クライアントを始動させる手段と、前記プロキシ・クライアントから前記第 1 の区分内の前記セキュリティ・サーバに前記要求を送る手段とをさらに含む、上記 ( 8 ) に記載の区分処理システム。

( 10 ) 前記主記憶域が前記第 1 の区分と前記第 2 の区分との間で共用される記憶域を含む、上記 ( 8 ) に記載の区分処理システム。

( 11 ) 前記第 1 の区分と前記第 2 の区分との間にリンクされた記憶域をさらに含み、前記セキュリティ・クライアントから前記許可要求を送る前記手段がメモリ間データ移動機構を含む、上記 ( 8 ) に記載の区分処理システム。

( 12 ) 前記プロキシ・クライアントから前記要求を送る前記手段が前記セキュリティ・サーバ固有のインタフェースを使用して前記プロキシ・クライアントによるプログラム呼出しを送る手段を含む、上記 ( 8 ) に記載の区分処理システム。

( 13 ) 前記共通セキュリティ・サーバから前記許可要求に対する応答を送る前記手段が、前記第 1 の区分内で稼働し、前記共通セキュリティ・サーバから前記セキュリティ・クライアントに前記応答を送る第 1 のプログラムをさらに含む、上記 ( 8 ) に記載の区分処理システム。

( 14 ) 第 1 の区分が共通セキュリティ・サーバを含み、第 2 の区分がセキュリティ・クライアントを含む区分処理システムにおいてセキュリティを設けるコンピュータ・プログラムであって、前記システムに、

a) ユーザによる許可要求を前記第 2 の区分内の前記セキュリティ・クライアントに送る手順と、

b) 前記許可要求を前記セキュリティ・クライアントから前記第 1 の区分内の前記共通セキュリティ・サーバに送る手順と、

c) 前記許可要求に対する第 1 の応答を前記第 1 の区分内の前記共通セキュリティ・サーバから前記第 2 の区分内の前記セキュリティ・クライアントに送るステップであって、区分間の前記要求または前記第 1 の応答のうちのいずれか一方の前記送信が主記憶域を経由する手順と、

d) 前記セキュリティ・クライアントから前記ユーザに第 2 の応答を送る手順とを実現させる、コンピュータ・プログラム。

( 15 ) 前記許可要求を送る手順が、

b 1) 前記第 2 の区分内で稼働する前記セキュリティ・クライアントが信号を送って前記第 1 の区分で稼働している第 1 のプログラムに前記第 1 の区分内のプロキシ・

クライアントを始動させる手順と、

b 2) 前記プロキシ・クライアントから前記第 1 の区分内の前記セキュリティ・サーバに前記要求を送る手順とをさらに含む、上記 ( 14 ) に記載のコンピュータ・プログラム。

( 16 ) 前記手順 b または手順 c において、前記第 1 の区分と少なくとも 1 つの前記第 2 の区分の 1 つとの間で共用される記憶域を使用する、上記 ( 14 ) に記載のコンピュータ・プログラム製品。

( 17 ) 前記手順 b または手順 c において、前記第 1 の区分と少なくとも 1 つの前記第 2 の区分の 1 つとの間にメモリ間データ移動機構によってリンクされた記憶域を使用する、上記 ( 14 ) に記載のコンピュータ・プログラム製品。

( 18 ) 前記手順 b において、前記セキュリティ・サーバ固有のインタフェースを使用して前記プロキシ・クライアントによるプログラム呼出しを供給する、上記 ( 14 ) に記載のコンピュータ・プログラム製品。

( 19 ) 前記手順 c において、前記共通セキュリティ・サーバから前記セキュリティ・クライアントに前記第 1 の区分内で稼働している第 1 のプログラムを介して前記応答を送る、上記 ( 14 ) に記載のコンピュータ・プログラム製品。

( 20 ) 共通セキュリティ・サーバを含む第 1 の区分とセキュリティ・クライアントを有する第 2 の区分とを有するセキュリティを設ける区分処理システムであって、前記第 1 の区分によるアクセスが可能な第 1 の部分と前記第 2 の区分によるアクセスが可能な第 2 の部分とを有する主記憶域と、前記セキュリティ・クライアントに接続され、ユーザによる許可要求を前記セキュリティ・クライアントに送る機構と、前記セキュリティ・クライアントから前記共通セキュリティ・サーバに前記許可要求を送る第 1 の送信部と、前記第 1 の区分内の前記共通セキュリティ・サーバから前記第 2 の区分内の前記セキュリティ・クライアントに前記許可要求に対する第 1 の応答を送る前記共通セキュリティ・サーバ内の第 2 の送信部と、を有し、前記区分間の前記要求または前記第 1 の応答のいずれか一方の送信が主記憶域を介し、さらに、前記セキュリティ・クライアントから前記ユーザに第 2 の応答を送る前記共通セキュリティ・サーバ内の第 3 の送信部とを含む、区分処理システム。

( 21 ) 前記第 1 の送信部が、プロキシ・クライアントを始動させる前記第 1 の区分内で稼働するプログラムと、前記プログラムに信号を送り、それによって前記第 1 の区分内の前記プロキシ・クライアントを始動させる前記セキュリティ・クライアントと、前記プロキシ・クライアントから前記セキュリティ・サーバに前記要求を送る第 4 の送信部とを含む、上記 ( 20 ) に記載の区分処理システム。

( 22 ) 前記主記憶域が、前記第 1 の区分と前記第 2 の

区分の両方によるアクセスが可能な第3の部分を含む、上記(20)に記載の区分処理システム。

(23) 前記第1の区分と前記第2の区分の間にリンクされた記憶域をさらに含み、前記第2の送信部が前記主記憶域の前記第1の部分と前記第2の部分との間でデータを移動するメモリ間データ移動機構を含む、上記(20)に記載の区分処理システム。

(24) 前記第4の送信部が、セキュリティ・サーバ固有のインタフェースを使用して前記プロキシ・クライアントによるプログラム呼出を送る、上記(21)に記載の区分処理システム。

(25) 前記第2の送信部が、前記第1の区分内で稼働し、前記共通セキュリティ・サーバから前記セキュリティ・クライアントに前記第1の応答を送るプログラムをさらに含む、上記(20)に記載の区分処理システム。

(26) 第2のセキュリティ・クライアントを有する第3の区分をさらに含み、前記第1の区分の前記共通セキュリティ・サーバが前記第2の区分内の前記セキュリティ・クライアントまたは前記第3の区分内の前記第2のセキュリティ・クライアントからの許可要求に応答する、上記(20)に記載の区分処理システム。

【図面の簡単な説明】

【図1】区分データ処理システムの概要を示す図である。

【図2】1つまたは複数のシステム・ボードから成る区分を有する物理区分処理システムを示す図である。

【図3】論理区分資源がそれぞれの区分に専用化された論理区分処理システムを示す図である。

【図4】論理区分資源を複数の区分間で動的に共用することができる、論理区分処理システムを示す図である。

【図5】UNIX(R)オペレーティング・システムの「プロセス間通信」の構造を示す図である。

【図6】スタンドアロン・ユーティリティによってロードされる構成テーブルに従って実メモリが共用される、本発明の一実施形態を示す図である。

【図7】入出力アダプタとそのドライバの機能を使用して区分間のデータ転送を容易にする、本発明の一実施形態を示す図である。

【図8】図7の実施形態の従来技術のシステムを示す図である。

【図9】区分間の実データ転送が、区分データ処理システムの通信ファブリックで実施されたデータ移動機構によって行われる、本発明の一実施形態を示す図である。

【図10】例示のデータ移動機構の構成要素を示す図である。

【図11】IBM S/390移動命令の例示の書式を示す図である。

【図12】アダプタ・データ移動を行う例示のステップを示す図である。

【図13】プロセッサ・データ移動を行う例示のステッ

プを示す図である。

【図14】ワークロード・マネージャ(WLM)の高水準図である。

【図15】典型的なワークロード・マネージャ・データを示す図である。

【図16】間接入出力を使用したクライアント/サーバのクラスタ化を示す図である。

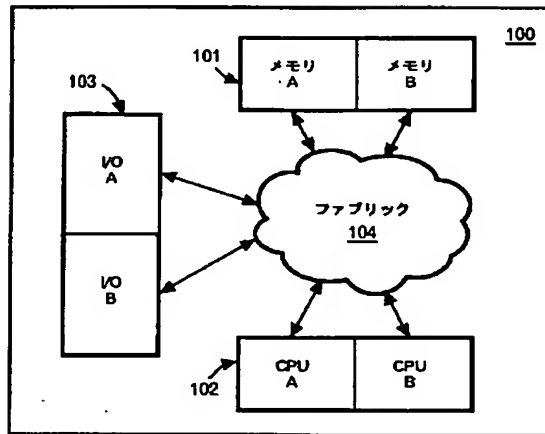
【図17】クライアント/サーバのサーバ・クラスタ化を示す図である。

【符号の説明】

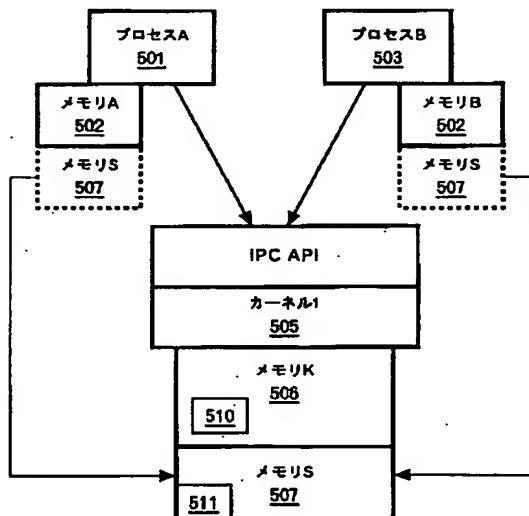
- 100 区分処理システム
- 101 メモリ資源ブロック
- 102 プロセッサ資源ブロック
- 103 入出力資源ブロック
- 104 相互接続ファブリック
- 200 物理区分処理システム
- 200A1 システム・ボード
- 201A メモリ
- 201B メモリ
- 202A プロセッサ
- 202B プロセッサ
- 203A 入出力装置
- 203B 入出力装置
- 204A 相互接続媒体
- 204B 相互接続媒体
- 205 相互接続ファブリック
- 400 論理区分資源共用システム
- 401 メモリ
- 402 プロセッサ
- 403 入出力資源
- 406 仮想プロセッサ
- 407 入出力ドライバ
- 408 ハイババイザ
- 501 プロセス
- 502 メモリ
- 503 プロセス
- 504 メモリ
- 505 カーネル
- 506 メモリ
- 507 メモリ
- 701 アプリケーション・プロセス
- 702 メモリ
- 703 プロセス
- 704 メモリ
- 705 カーネル1
- 706 カーネル・メモリ1
- 707 カーネル・メモリ2
- 708 メモリK2
- 716 デバイス・ドライバ
- 717 デバイス・ドライバ

718 ソケットAPI  
 719 ソケットAPI  
 720 入出力アダプタ  
 801 プロセス  
 802 メモリ  
 803 プロセス  
 804 メモリ  
 805 カーネル  
 806 メモリ

【図1】

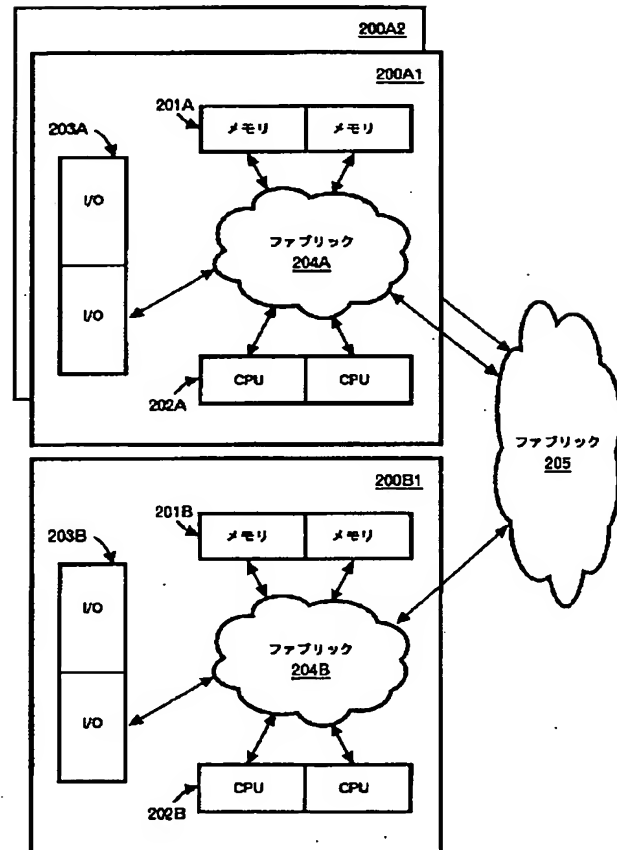


【図5】



807 カーネル  
 808 メモリ  
 816 デバイス・ドライバ  
 817 デバイス・ドライバ  
 818 ソケットAPI  
 819 ソケットAPI  
 820 入出力アダプタ  
 821 データ移動機構

【図2】



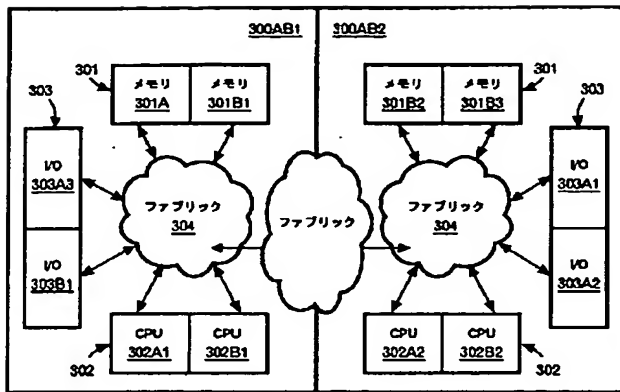
【図11】

1000

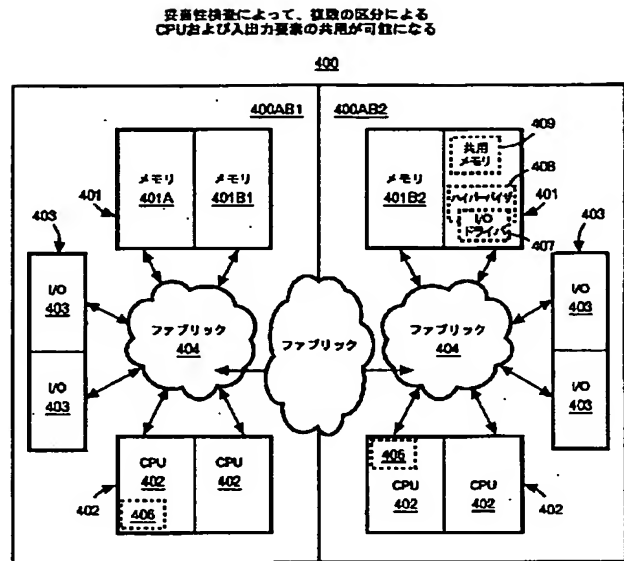
MVXL	カウント	宛先 n	ソース
------	------	------	-----

MVXLが、ソース・レジスタによって指定された物理アドレスから宛先レジスタによって指定された物理アドレスに、カウント・レジスタによって指定されたバイト数を移動する  
 この命令は特権命令である  
 (MVCLが、仮想アドレス間でこれと同じ機能を実行する)  
 ここでは、デバイス・ドライバはレジスタに仮想ドライバではなく物理アドレスをロードし、それによって区分間データ移動を可能にする

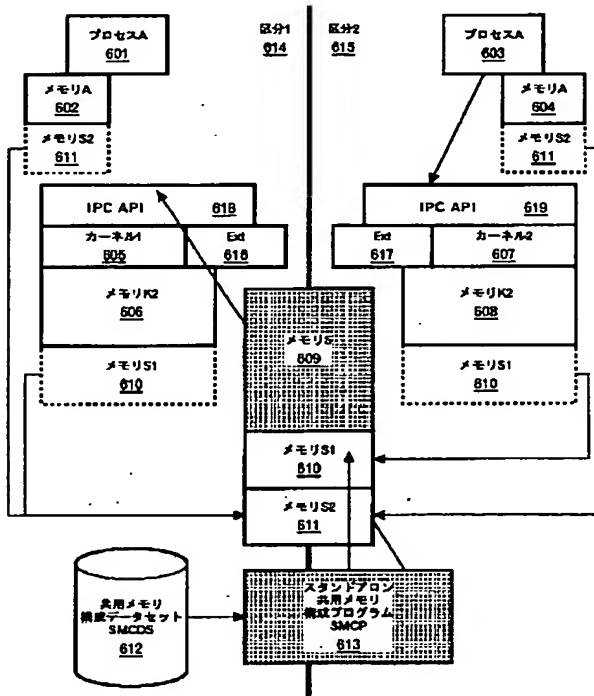
【図 3】



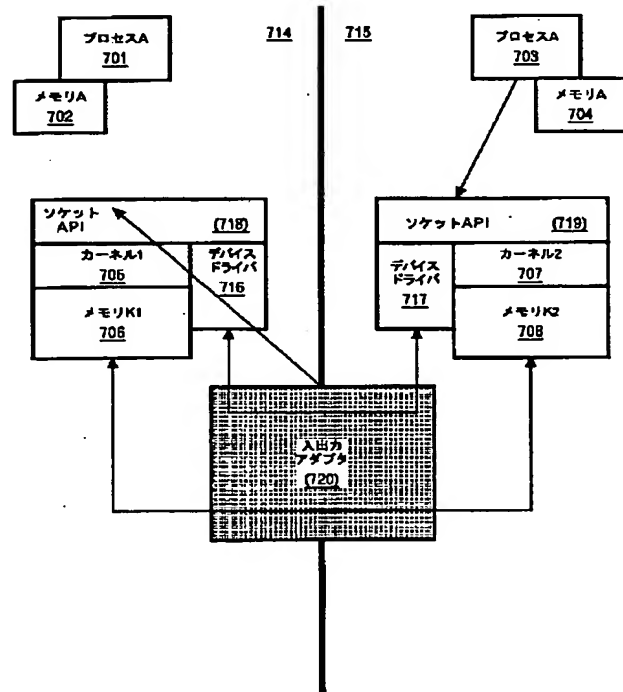
【図 4】



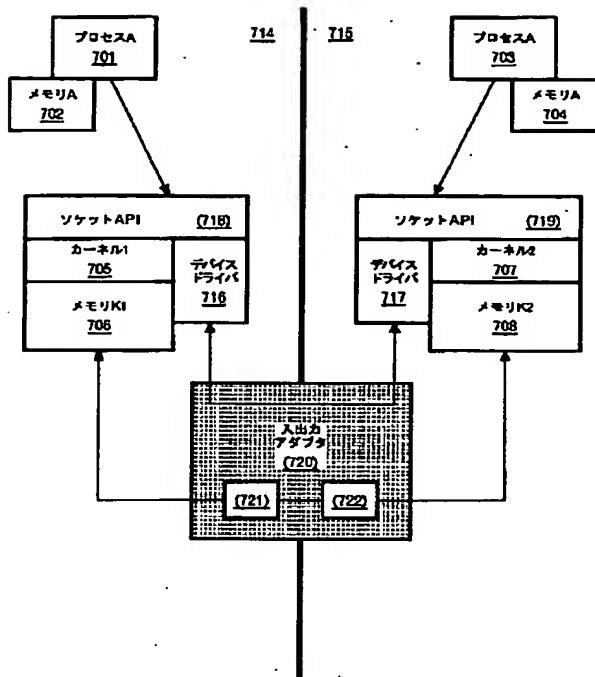
【図 6】



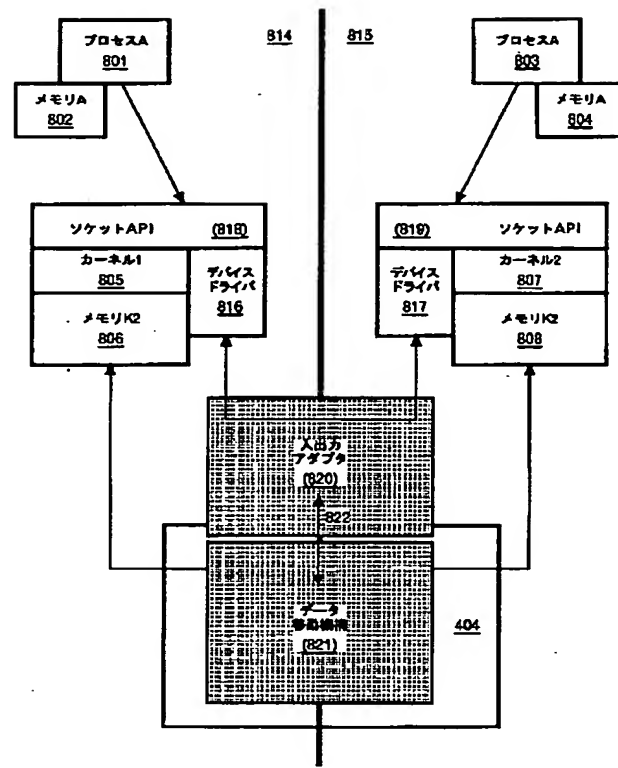
【図 7】



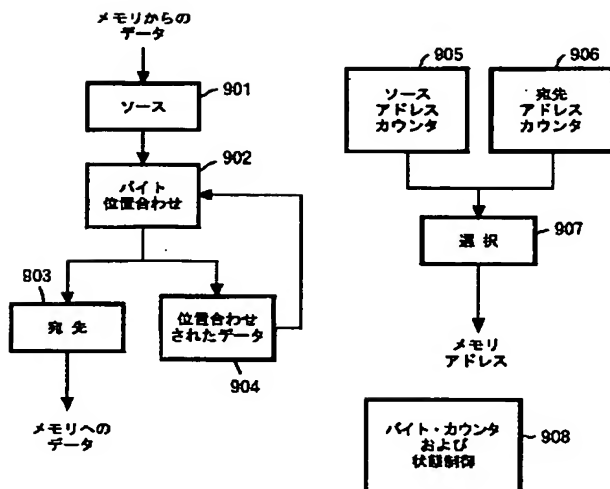
【図 8】



【図 9】



【図 10】



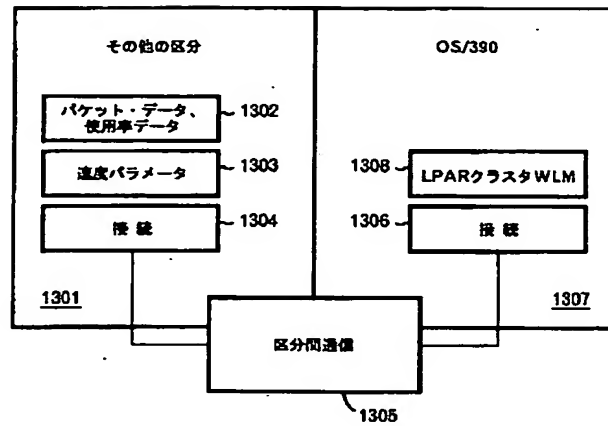
【図 12】

1101. ユーザがデバイス・ドライバを呼び出す  
 -以下を提供する  
 ソース・ネットワークID  
 ソース・オフセット  
 宛先ネットワークID
1102. デバイス・ドライバがアドレスをアダプタに転送する
1103. アダプタがアドレスを変換する  
 -IDから物理基座アドレスを探索する(テーブル・ルックアップ)  
 -ロックおよび現行宛先オフセットを入手する  
 -オフセットを加える  
 -境界を調べる
1104. アダプタがカウンタとアドレスをレジスタにロードする
1105. アダプタがデータ移動を実行する
1106. アダプタがロックを解放する
1107. アダプタがデバイス・ドライバに通知し、デバイス・ドライバがユーザに「戻る」

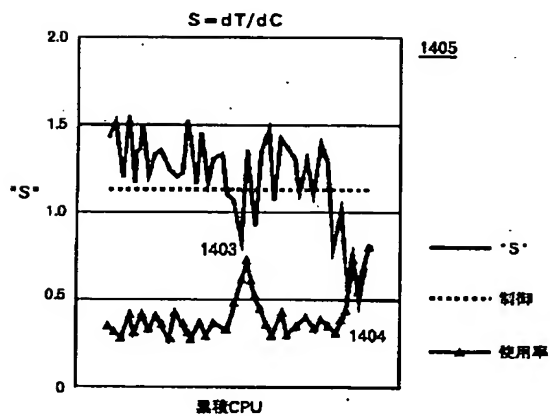
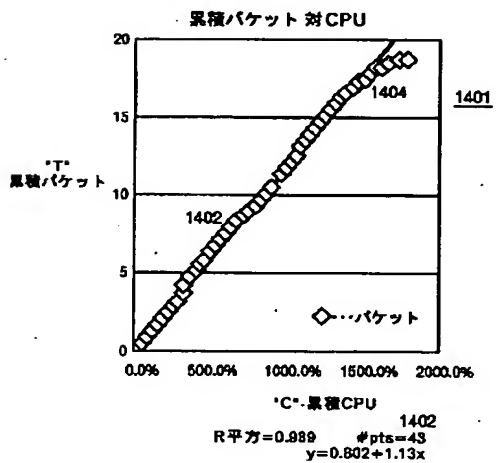
【図 13】

1201. ユーザがデバイス・ドライバを呼び出す  
 -以下を供給する  
   ソース・ネットワークID  
   ソース・オフセット  
   宛先ネットワークID
1202. デバイス・ドライバがアダプタにアドレスを送る
1203. アダプタが変換する  
 -IDから物理基座アドレスを探索する(テーブル・ルックアップ)  
 -ロックおよび実行優先オフセットを入手する  
 -オフセットを加える  
 -境界を調べる  
 -ロックと物理アドレスをデバイス・ドライバに返す
1204. デバイス・ドライバがデータ移動を実行する
1205. デバイス・ドライバがロックを解放する
1206. デバイス・ドライバが戻る

【図 14】



【図 15】



【図 16】

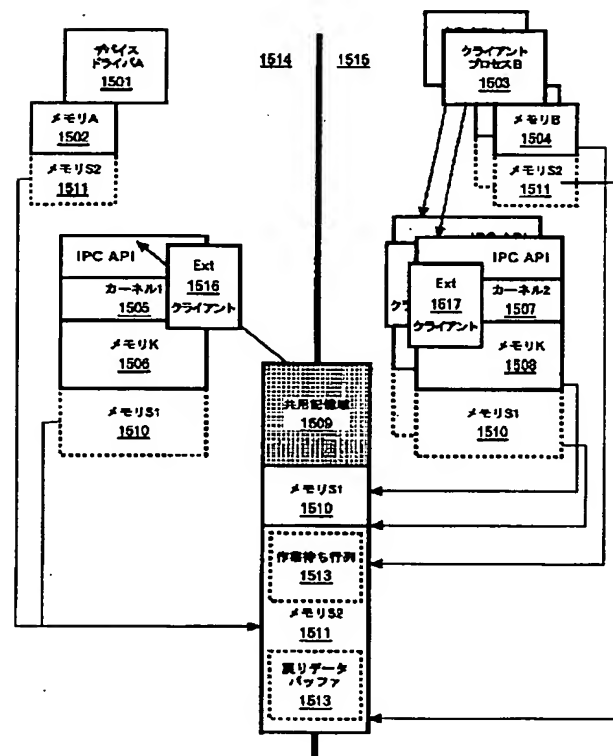


Figure 1 is a schematic diagram of a system architecture. It shows two main components, 1814 and 1815, connected by a central vertical line. Component 1814 (left) includes a 'セキエリディ ユーザ1A' (1601), 'メモリA' (1602), 'メモリS2' (1611), 'IPC API' (1605), 'カーネル1' (1606), 'メモリK' (1610), and 'メモリS1' (1610). Component 1815 (right) includes a 'クライアントプロセスB' (1603), 'メモリB' (1604), 'メモリS2' (1611), 'IPC API' (1607), 'Ext' (1617), 'カーネル2' (1608), 'メモリK' (1610), and 'メモリS1' (1610). A central vertical line separates the two components. At the bottom, a stack of components is shown: '汎用記憶域' (1609), 'メモリS1' (1610), '作業待ち行先' (1613), 'メモリS2' (1611), and '読みデータ バッファ' (1613). A user icon is shown at the top right, connected to component 1815.

(72) 発明者    マイケル・イー・バスキー  
                  アメリカ合衆国12590    ニューヨーク州ワ  
                  ピンジャーズ・フォールズ    ハイビュー・  
                  ロード    31

(72) 発明者    フランク・ジェイ・デジリオ  
                  アメリカ合衆国12603-4621    ニューヨー  
                  ク州ポーキプシー    ハイ・リッジ・ロード  
                  13

(72) 発明者    ジョン・シィ・ジョーンズ  
                  アメリカ合衆国30062    ジョージア州マリ  
                  エッタ    ジョーダン・レーク・ドライブ  
                  4010

(72)発明者 クリスチャン・エフ・ローバツハ  
アメリカ合衆国12603 ニューヨーク州ボ  
ーキプシー オーバールック・ロード  
191

(72)発明者 ジョセフ・エル・テンブル・ザ・サード  
アメリカ合衆国12443 ニューヨーク州ハ  
ーリー フック・ストリート 312 ビ  
ー・オー・ボックス 507

Fターム(参考) 5B045 BB28 BB42 GG01  
5B085 AE00 BA06 BG07  
5B089 GB01 JB14 KA17 KB13  
5B098 HH04